Package 'spatempBFA'

November 21, 2024

Type Package

Title Spatiotemporal Bayesian Factor Analysis with Enhanced Scalability

Version 1.3

Author Yifan Cheng

Maintainer Yifan Cheng <y.cheng@u.nus.edu>

Description This repository implements practicable Bayesian nonparametric factor models capable of tackling moderately large spatiotemporal data along with subsequent vital inferential tasks including spatial and temporal prediction as well as spatial clustering of temporal trends. Details pertaining to our methods are in 'Enhancing Scalability in Bayesian Nonparametric Factor Analysis of Spatiotemporal Data' <arXiv:2312.05802>, which was submitted to Journal of Computational and Graphical Statistics.

License GPL (>= 2)

Encoding UTF-8

LazyData true

LazyDataCompression xz

NeedsCompilation yes

Depends R

Imports Rcpp (>= 1.0.9), graphics, grDevices, msm (>= 1.0.0), mvtnorm (>= 1.0-0), tmvtnorm, pgdraw (>= 1.0), stats, utils

LinkingTo Rcpp, RcppArmadillo

Suggests coda, classInt, knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.2

R topics documented:

`bfaFixedL` *Spatial factor analysis using a Bayesian hierarchical model.*

Description

`bfaFixedL` is a Markov chain Monte Carlo (MCMC) sampler for a Bayesian spatial factor analysis model. The spatial component is introduced using a Probit stick-breaking process prior on the factor loadings. The model is implemented using a Bayesian hierarchical framework.

Usage

```
bfaFixedL(
  formula,
  data,
  dist,
  time,
  K,
  L = 20,
  family = "normal",
  temporal.structure = "exponential",
  spatial.structure = "continuous",
  starting = NULL,
  hypers = NULL,
  tuning = NULL,
  mcmc = NULL,
  seed = 27,
  gamma.shrinkage = TRUE,
  include.time = TRUE,
  include.space = TRUE,
  clustering = TRUE,
  seasonPeriod = 1,
  equalTimeDist = TRUE,
  spatApprox = TRUE,
  alphaMethod = "block",
  h = 15,
  storeSpatPredPara = TRUE,
  storeWeights = TRUE,
  alphasWeightsToFiles = FALSE
)
```

Arguments

formula	A formula object, corresponding to the spatial factor analysis model. The response must be on the left of a ~ operator, and the terms on the right must indicate the covariates to be included in the fixed effects. If no covariates are desired a zero should be used, ~ 0.
data	A required data.frame containing the variables (Y, additional x coviariate(s) if there is/are, and a trials variable if there is/are observation type(s) from family "binomial") in the model. The data frame must contain M x O x Nu rows. Here, M represents the number of spatial locations, O the number of different observation types and Nu the number of temporal visits. The observations must be first ordered spatially, second by observation type and then temporally. This means that the first M x O observations come from the first time point and the first M observations come the first spatial observation type. If there is/are observation type(s) from family "binomial", then trials contains the numbers of trials as positive integers for each of the binomial observations. Entries in trials corresponding to non-binomial data can be specified as any arbitrary positive integer. The function will change these values to 1.
dist	A M x M dimensional distance matrix. For a discrete spatial process the matrix contains binary adjacencies that dictate the spatial neighborhood structure and for continuous spatial processes the matrix should be a continuous distance matrix (e.g., Euclidean).
time	A Nu dimensional vector containing the observed time points in increasing order.
K	A scalar that indicates the dimension (i.e., quantity) of latent factors.
L	A positive integer indicating the number of latent clusters for each column of the factor loadings matrix.
family	A character string or a vector of length O (if O > 1) of character strings indicating the distribution(s) of the observed data. Options for each observation type include: "normal", "probit", "tobit", and "binomial". If O > 1 and family is of length 1, then all of the O observation types are from the same family (distribution). Any combination of likelihoods can be used.
temporal.structure	Character string indicating the temporal kernel. Options include: "exponential", "ar1", "sar1", and "sexponential".
spatial.structure	Character string indicating the type of spatial process. Options include: "continuous" (i.e., Gaussian process with exponential kernel) and "discrete" (i.e., proper CAR).
starting	Either NULL or a list containing starting values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the starting values may be specified. When NULL is chosen then default starting values are automatically generated. Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon or Psi containing appropriate objects. Beta (or Delta) must either be a P (or K) dimensional vector or a scalar (the scalar populates the entire vector). Sigma2 must be either a M x (O - C) matrix or a scalar. Kappa must be a O x O dimensional matrix, Rho a scalar, Upsilon a K x K matrix, and Psi a scalar.
hypers	Either NULL or a list containing hyperparameter values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the hyperparameter values may be specified.

When NULL is chosen then default hyperparameter values are automatically generated. These default hyperparameters are described in detail in (Berchuck et al.). Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon or Psi containing further hyperparameter information. These objects are themselves lists and may be constructed as follows.

Beta is a list with two objects, MuBeta and SigmaBeta. These values represent the prior mean and variance parameters for the multivariate normal prior.

Delta is a list with two objects, A1 and A2. These values represent the prior shape parameters for the multiplicative Gamma shrinkage prior.

Sigma2 is a list with two objects, A and B. These values represent the shape and scale for the variance parameters.

Kappa is a list with two objects, SmallUpsilon and BigTheta. SmallUpsilon represents the degrees of freedom parameter for the inverse-Wishart hyperprior and must be a real number scalar, while BigTheta represents the scale matrix and must be a O x O dimensional positive definite matrix.

Rho is a list with two objects, ARho and BRho. ARho represents the lower bound for the uniform hyperprior, while BRho represents the upper bound. The bounds must be specified carefully. This is only specified for continuous spatial processes.

Upsilon is a list with two objects, Zeta and Omega. Zeta represents the degrees of freedom parameter for the inverse-Wishart hyperprior and must be a real number scalar, while Omega represents the scale matrix and must be a K x K dimensional positive definite matrix.

Psi is a list with two objects, dependent on if the temporal kernel is exponential or ar1. For exponential, the two objects are APsi and BPsi. APsi represents the lower bound for the uniform hyperprior, while BPsi represents the upper bound. The bounds must be specified carefully. For ar1, the two objects are Beta and Gamma, which are the two shape parameters of a Beta distribution shifted to have domain in (-1, 1).

tuning — Either NULL or a list containing tuning values to be specified for the MCMC Metropolis steps. If NULL is not chosen then all of the tuning values must be specified.

When NULL is chosen then default tuning values are automatically generated to 1. Otherwise a list must be provided with names Psi, or Rho. Each of these entries must be scalars containing tuning variances for their corresponding Metropolis updates.

mcmc — Either NULL or a list containing input values to be used for implementing the MCMC sampler. If NULL is not chosen then all of the MCMC input values must be specified.

NBurn: The number of sampler scans included in the burn-in phase. (default = 10,000)

NSims: The number of post-burn-in scans for which to perform the sampler. (default = 10,000)

NThin: Value such that during the post-burn-in phase, only every NThin-th scan is recorded for use in posterior inference (For return values we define, NKeep = NSims / NThin (default = 1).

NPilot: The number of times during the burn-in phase that pilot adaptation is performed (default = 20)

seed — An integer value used to set the seed for the random number generator

gamma.shrinkage
: A logical indicating whether a gamma shrinkage process prior is used for the variances of the factor loadings columns. If FALSE, the hyperparameters (A1 and A2) indicate the shape and rate for a gamma prior on the precisions. Default is TRUE. It can only be TRUE when clustering = TRUE.

include.time
: A logical indicating whether a temporal process should be included. Default is TRUE, however if FALSE the temporal correlation structure matrix H(\psi) is fixed as the T x T identity matrix. This specification overrides all inputs temporal.structure, equalTimeDist, and seasonPeriod.

include.space
: A logical indicating whether a spatial process should be included. Default is TRUE, however if FALSE the spatial correlation matrix F(\rho) is fixed as the M x M identity matrix. This specification overrides the inputs spatial.structure, spatApprox, and alphaMethod.

clustering
: A logical indicating whether the Bayesian non-parametric process should be used, default is TRUE. If FALSE is specified each column is instead modeled with an independent spatial process.

seasonPeriod
: A positive integer value indicating the temporal seasonality period. Default is 1 (no temporal seasonality). This argument is only used (under which scenario it should be a positive integer greater than 1) when include.time = TRUE and the temporal.structure input corresponds to a structure with seasonality.

equalTimeDist
: A logical indicating whether the distances between adjacent time points are equal. Default is TRUE. If TRUE, the user should normalize the time distance to 1 (for the time input).

spatApprox
: A logical indicating whether spatial nearest neighbor kriging is used for the latent location-specific spatial parameter vectors \alpha_{jl_j}'s to accelerate computation. Default is TRUE. When TRUE, spatial.structure must be "continuous". If FALSE, the exact M x M neighborhood structure matrix F(\rho) is adopted throughout.

alphaMethod
: Character string indicating which approach is used to update the \alpha_{jl_j}'s in the Gibbs Sampler. Options include: "block" and "sequential". This argument is only used when include.space = TRUE and spatApprox = TRUE.

h
: a positive integer much smaller than M, the number of location points, indicating an upper bound of the number of nearest neighbors for each location point.

storeSpatPredPara
: a logical indicating whether we want to store the posterior samples for Xi, Delta, Tau, Alpha, Theta when clustering = TRUE. Has to be TRUE if we want to perform predictions at new spatial location(s). Can be FALSE to save storage space if spatial prediction is not required.

storeWeights
: a logical indicating whether we want to store the posterior Weights estimates when clustering = TRUE. Has to be TRUE if we want to perform temporal trends clustering from the output object. Can be FALSE to save storage space if otherwise.

alphasWeightsToFiles
: a logical indicating whether we want to store the posterior alpha and weights estimates to files (one for each kept MCMC iteration) instead of including them in the main output list. Can only be TRUE if clustering = TRUE and at least one of storeSpatPredPara and storeWeights is TRUE. When that is the case, specifying alphasWeightsToFiles = TRUE would be immensely useful if we don't have enough space to output a model fit object with all kept posterior parameter estimates. Default is FALSE.

Value

bfaFixedL returns a list containing the following objects (some may be NULL)

lambda NKeep x (M x O x K) matrix of posterior samples for the factor loadings matrix Lambda. The labels for each column are Lambda_O_M_K.

eta NKeep x (Nu x K) matrix of posterior samples for the latent factors eta. The labels for each column are Eta_Nu_K.

beta NKeep x P matrix of posterior samples for beta.

sigma2 NKeep x (M * (O - C)) matrix of posterior samples for the variances sigma2. The labels for each column are Sigma2_O_M.

kappa NKeep x ((O * (O + 1)) / 2) matrix of posterior samples for kappa. The columns have names that describe the samples within them. The row is listed first, e.g., Kappa3_2 refers to the entry in row 3, column 2.

delta NKeep x K matrix of posterior samples for delta.

tau NKeep x K matrix of posterior samples for tau.

upsilon NKeep x ((K * (K + 1)) / 2) matrix of posterior samples for Upsilon. The columns have names that describe the samples within them. The row is listed first, e.g., Upsilon3_2 refers to the entry in row 3, column 2.

psi NKeep x 1 matrix of posterior samples for psi.

xi NKeep x (M x O x K) matrix of posterior samples for factor loadings cluster labels xi. The labels for each column are Xi_O_M_K.

rho NKeep x 1 matrix of posterior samples for rho.

theta NKeep x (L x K) matrix of posterior samples for theta.

alpha NKeep x (M x O x K x (L - 1) matrix of posterior samples for alpha. For each kept row (MCMC iteration) in alpha, the corresponding entries are ordered first by observation type, then spatially, then by clustering group, and finally by factor.

weights NKeep x (M x O x K x L) matrix of posterior samples for weights. For each kept row (MCMC iteration) in weights, the corresponding entries are ordered first spatially, then by observation type, then by clustering group, and finally by factor.

GibbsStepTime NKeep x 10 or NKeep x 6 matrix of posterior sampling computation time for Gibbs sampler steps corresponding to parameters except beta and sigma2.

metropolis 2 (or 1) x 3 matrix of metropolis acceptance rates, updated tuners, and original tuners that result from the pilot adaptation.

datobj A list of data objects that are used in future bfaFixedL functions and should be ignored by the user.

dataug A list of data augmentation objects that are used in future bfaFixedL functions and should be ignored by the user.

runtime A character string giving the runtime of the MCMC sampler.

bfaVaryingLjs *Spatial factor analysis using a Bayesian hierarchical model.*

Description

bfaVaryingLjs is a Markov chain Monte Carlo (MCMC) sampler for a Bayesian spatial factor analysis model. The spatial component is introduced using a Probit stick-breaking process prior on the factor loadings. The model is implemented using a Bayesian hierarchical framework.

Usage

```
bfaVaryingLjs(
  formula,
  data,
  dist,
  time,
  K,
  LjVec,
  family = "normal",
  temporal.structure = "exponential",
  spatial.structure = "continuous",
  starting = NULL,
  hypers = NULL,
  tuning = NULL,
  mcmc = NULL,
  seed = 27,
  gamma.shrinkage = TRUE,
  include.time = TRUE,
  include.space = TRUE,
  seasonPeriod = 1,
  equalTimeDist = TRUE,
  spatApprox = TRUE,
  alphaSequen = FALSE,
  h = 15,
  storeSpatPredPara = TRUE,
  storeWeights = TRUE
)
```

Arguments

formula	A formula object, corresponding to the spatial factor analysis model. The response must be on the left of a ~ operator, and the terms on the right must indicate the covariates to be included in the fixed effects. If no covariates are desired a zero should be used, ~ 0.
data	A required data.frame containing the variables (Y, additional x coviariate(s) if there is/are, and a trials variable if there is/are observation type(s) from family "binomial") in the model. The data frame must contain M x O x Nu rows. Here, M represents the number of spatial locations, O the number of different observation types and Nu the number of temporal visits. The observations must be first ordered spatially, second by observation type and then temporally. This means that the first M x O observations come from the first time point and

the first M observations come the first spatial observation type. If there is/are observation type(s) from family "binomial", then trials contains the numbers of trials as positive integers for each of the binomial observations. Entries in trials corresponding to non-binomial data can be specified as any arbitrary positive integer. The function will change these values to 1.

dist — A M x M dimensional distance matrix. For a discrete spatial process the matrix contains binary adjacencies that dictate the spatial neighborhood structure and for continuous spatial processes the matrix should be a continuous distance matrix (e.g., Euclidean).

time — A Nu dimensional vector containing the observed time points in increasing order.

K — A scalar that indicates the dimension (i.e., quantity) of latent factors.

LjVec — A vector of length K consisting of positive integers indicating the starting numbers of latent clusters for the K columns of the factor loadings matrix.

family — A character string or a vector of length O (if O > 1) of character strings indicating the distribution(s) of the observed data. Options for each observation type include: "normal", "probit", "tobit", and "binomial". If O > 1 and family is of length 1, then all of the O observation types are from the same family (distribution). Any combination of likelihoods can be used.

temporal.structure — Character string indicating the temporal kernel. Options include: "exponential", "ar1", "sar1", and "sexponential".

spatial.structure — Character string indicating the type of spatial process. Options include: "continuous" (i.e., Gaussian process with exponential kernel) and "discrete" (i.e., proper CAR).

starting — Either NULL or a list containing starting values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the starting values may be specified.

When NULL is chosen then default starting values are automatically generated. Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon or Psi containing appropriate objects. Beta (or Delta) must either be a P (or K) dimensional vector or a scalar (the scalar populates the entire vector). Sigma2 must be either a M x (O - C) matrix or a scalar. Kappa must be a O x O dimensional matrix, Rho a scalar, Upsilon a K x K matrix, and Psi a scalar.

hypers — Either NULL or a list containing hyperparameter values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the hyperparameter values may be specified.

When NULL is chosen then default hyperparameter values are automatically generated. These default hyperparameters are described in detail in (Berchuck et al.). Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon or Psi containing further hyperparameter information. These objects are themselves lists and may be constructed as follows.

Beta is a list with two objects, MuBeta and SigmaBeta. These values represent the prior mean and variance parameters for the multivariate normal prior.

Delta is a list with two objects, A1 and A2. These values represent the prior shape parameters for the multiplicative Gamma shrinkage prior.

Sigma2 is a list with two objects, A and B. These values represent the shape and scale for the variance parameters.

Kappa is a list with two objects, SmallUpsilon and BigTheta. SmallUpsilon represents the degrees of freedom parameter for the inverse-Wishart hyperprior

and must be a real number scalar, while BigTheta represents the scale matrix and must be a O x O dimensional positive definite matrix.

Rho is a list with two objects, ARho and BRho. ARho represents the lower bound for the uniform hyperprior, while BRho represents the upper bound. The bounds must be specified carefully. This is only specified for continuous spatial processes.

Upsilon is a list with two objects, Zeta and Omega. Zeta represents the degrees of freedom parameter for the inverse-Wishart hyperprior and must be a real number scalar, while Omega represents the scale matrix and must be a K x K dimensional positive definite matrix.

Psi is a list with two objects, dependent on if the temporal kernel is exponential or ar1. For exponential, the two objects are APsi and BPsi. APsi represents the lower bound for the uniform hyperprior, while BPsi represents the upper bound. The bounds must be specified carefully. For ar1, the two objects are Beta and Gamma, which are the two shape parameters of a Beta distribution shifted to have domain in (-1, 1).

tuning — Either NULL or a list containing tuning values to be specified for the MCMC Metropolis steps. If NULL is not chosen then all of the tuning values must be specified.

When NULL is chosen then default tuning values are automatically generated to 1. Otherwise a list must be provided with names Psi, or Rho. Each of these entries must be scalars containing tuning variances for their corresponding Metropolis updates.

mcmc — Either NULL or a list containing input values to be used for implementing the MCMC sampler. If NULL is not chosen then all of the MCMC input values must be specified.

NBurn: The number of sampler scans included in the burn-in phase. (default = 10,000)

NSims: The number of post-burn-in scans for which to perform the sampler. (default = 10,000)

NThin: Value such that during the post-burn-in phase, only every NThin-th scan is recorded for use in posterior inference (For return values we define, NKeep = NSims / NThin (default = 1).

NPilot: The number of times during the burn-in phase that pilot adaptation is performed (default = 20)

seed — An integer value used to set the seed for the random number generator

gamma.shrinkage

A logical indicating whether a gamma shrinkage process prior is used for the variances of the factor loadings columns. If FALSE, the hyperparameters (A1 and A2) indicate the shape and rate for a gamma prior on the precisions. Default is TRUE.

include.time — A logical indicating whether a temporal process should be included. Default is TRUE, however if FALSE the temporal correlation structure matrix H(\psi) is fixed as the T x T identity matrix. This specification overrides all inputs temporal.structure, equalTimeDist, and seasonPeriod.

include.space — A logical indicating whether a spatial process should be included. Default is TRUE, however if FALSE the spatial correlation matrix F(\rho) is fixed as the M x M identity matrix. This specification overrides the inputs spatial.structure, spatApprox, and alphaMethod.

seasonPeriod A positive integer value indicating the temporal seasonality period. Default is 1 (no temporal seasonality). This argument is only used (under which scenario it should be a positive integer greater than 1) when include.time = TRUE and the temporal.structure input corresponds to a structure with seasonality.

equalTimeDist A logical indicating whether the distances between adjacent time points are equal. Default is TRUE. If TRUE, the user should normalize the time distance to 1 (for the time input).

spatApprox A logical indicating whether spatial nearest neighbor kriging is used for the latent location-specific spatial parameter vectors \alpha_{jl_j}'s to accelerate computation. Default is TRUE. When TRUE, spatial.structure must be "continuous". If FALSE, the exact M x M neighborhood structure matrix F(\rho) is adopted throughout.

alphaSequen A logical indicating whether we want to sequentially update the alpha's. This argument is only used when include.space = TRUE and spatApprox = TRUE.

h a positive integer much smaller than M, the number of location points, indicating an upper bound of the number of nearest neighbors for each location point.

storeSpatPredPara a logical indicating whether we want to store the posterior samples for Alpha, Theta. Has to be TRUE if we want to perform predictions at new spatial location(s). Can be FALSE to save storage space if spatial prediction is not required.

storeWeights a logical indicating whether we want to store the posterior Weights estimates. Has to be TRUE if we want to perform temporal trends clustering from the output object. Can be FALSE to save storage space if otherwise.

Value

bfaVaryingLjs returns a list containing the following objects (some may be NULL)

lambda NKeep x (M x O x K) matrix of posterior samples for the factor loadings matrix Lambda. The labels for each column are Lambda_O_M_K.

eta NKeep x (Nu x K) matrix of posterior samples for the latent factors eta. The labels for each column are Eta_Nu_K.

beta NKeep x P matrix of posterior samples for beta.

sigma2 NKeep x (M * (O - C)) matrix of posterior samples for the variances sigma2. The labels for each column are Sigma2_O_M.

kappa NKeep x ((O * (O + 1)) / 2) matrix of posterior samples for kappa. The columns have names that describe the samples within them. The row is listed first, e.g., Kappa3_2 refers to the entry in row 3, column 2.

delta NKeep x K matrix of posterior samples for delta.

tau NKeep x K matrix of posterior samples for tau.

upsilon NKeep x ((K * (K + 1)) / 2) matrix of posterior samples for Upsilon. The columns have names that describe the samples within them. The row is listed first, e.g., Upsilon3_2 refers to the entry in row 3, column 2.

psi NKeep x 1 matrix of posterior samples for psi.

xi NKeep x (M x O x K) matrix of posterior samples for factor loadings cluster labels xi. The labels for each column are Xi_O_M_K.

rho NKeep x 1 matrix of posterior samples for rho.

ljvec NKeep x K matrix of posterior samples for Lj's.

`theta` A `list` of posterior samples for `theta`. The list contains `K x NKeep` elements (ordered by column), each of which is a vector of length `Lj` at that MCMC iteration.

`alpha` A `list` of posterior samples for `alpha`. The list contains `K x NKeep` elements (ordered by column, i.e., the first K components of the list correspond to the first kept iteration and so on), each of which is a `matrix` of dimension `(Lj - 1) x (M x O)` for `Lj` at that MCMC iteration. If `Lj = 1` for a certain `j` at a particular MCMC iteration, then that corresponding `matrix` in `alpha` is of dimension `1 x (M x O)` with entries all set to +Inf. In each element matrix for each clustering group, the corresponding entries are ordered first by observation type, then spatially. (the first O correspond to the first location point, the next O correspond to the second location point and so on)

`weights` A `list` of posterior samples for `weights`. The list contains `K x NKeep` elements (ordered by column), each of which is a `matrix` of dimension `Lj x (M x O)` for `Lj` at that MCMC iteration. In each element matrix for each clustering group, the corresponding entries are ordered first spatially, then by observation type. (the first M correspond to the first observation type, the next M correspond to the second observation type and so on)

`GibbsStepTime` `NKeep x 10 matrix` of posterior sampling computation time for Gibbs sampler steps corresponding to parameters except `beta` and `sigma2`.

`metropolis` `2 (or 1) x 3 matrix` of metropolis acceptance rates, updated tuners, and original tuners that result from the pilot adaptation.

`datobj` A `list` of data objects that are used in future `bfaVaryingLjs` functions and should be ignored by the user.

`dataug` A `list` of data augmentation objects that are used in future `bfaVaryingLjs` functions and should be ignored by the user.

`runtime` A `character` string giving the runtime of the MCMC sampler.

`clusteringFixedL` *clusteringFixedL*

Description

Cluster spatial points into regions with similar temporal trajectories from a `FixedLbfa` model.

Usage

```
clusteringFixedL(object, o = 1, nkeep = 1, nCent = 3)
```

Arguments

`object`	A `FixedLbfa` or `FixedLbfaVAR1` model object on which sptial temporal trends clustering is based.
`o`	A positive integer less than or equal to O (the number of observation types) indicating the single observation type that we want to perform clustering for.
`nkeep`	A positive integer less than or equal to NKeep (the number of post-burn-in post-thinned MCMC iterations) indicating the number of selected kept post-burn-in MCMC iterations whose posterior parameter estimates our spatial temporal trends clustering will be based on. If `nkeep = 0`, the posterior mean weights estimates will be used for our clustering analysis.
`nCent`	A positive integer indicating the number of clusters for k-means clustering.

Value

clusteringFixedL returns an object from R's kmeans() function, which is a list with elements cluster, centers, totss, withinss, tot.withinss, betweenss, size, iter, and ifault.

References

Yifan Cheng

---

clusteringVaryLj *clusteringVaryLj*

Description

Cluster spatial points into regions with similar temporal trajectories from a varyLjBFA model.

Usage

```
clusteringVaryLj(object, o = 1, nkeep = 1, nCent = 3)
```

Arguments

object	A varyLjBFA or VAR1varyLjBFA model object on which sptial temporal trends clustering is based.
o	A positive integer less than or equal to O (the number of observation types) indicating the single observation type that we want to perform clustering for.
nkeep	A positive integer less than or equal to NKeep (the number of post-burn-in post-thinned MCMC iterations) indicating the number of selected kept post-burn-in MCMC iterations whose posterior parameter estimates our spatial temporal trends clustering will be based on.
nCent	A positive integer indicating the number of clusters for k-means clustering.

Value

clusteringVaryLj returns an object from R's kmeans() function, which is a list with elements cluster, centers, totss, withinss, tot.withinss, betweenss, size, iter, and ifault.

References

Yifan Cheng

diagnostics *diagnostics*

Description

Calculates diagnostic metrics based on outputs from a FixedLbfa or varyLjBFA model.

Usage

```
diagnostics(
  object,
  diags = c("dic", "dinf", "meanIC", "waic"),
  keepDeviance = FALSE,
  keepPPD = FALSE,
  Verbose = TRUE,
  seed = 29
)
```

Arguments

object	A FixedLbfa or varyLjBFA or FixedLbfaVAR1 or VAR1varyLjBFA model object for which diagnostics are desired from.
diags	A vector of character strings indicating which diagnostics to compute. Options include: Deviance Information Criterion ("dic"), d-infinity ("dinf"), Posterior Mean Information Criteria ("meanIC"), and Watanabe-Akaike Information Criterion ("waic"). At least one option must be included. Note: The probit model cannot compute the DIC or WAIC diagnostics due to computational issues regarding the multivariate normal CDF.
keepDeviance	A logical indicating whether the posterior deviance distribution is returned (default = FALSE).
keepPPD	A logical indicating whether the posterior predictive distribution at each observed location is returned (default = FALSE).
Verbose	A boolean logical indicating whether progress should be output (default = TRUE).
seed	An integer value used to set a seed for the random number generator (default = 54).

Details

To assess model fit, DIC, d-infinity, meanIC, and WAIC are utilized. DIC is based on the deviance statistic and penalizes the complexity of a model with an effective number of parameters estimate pD (Spiegelhalter et al 2002). The d-infinity posterior predictive measure is an alternative diagnostic tool, where d-infinity=P+G. The G term decreases as goodness of fit increases, and P, the penalty term, inflates as the model becomes over-fit. Hence, small values of both of these terms and thus small values of d-infinity are desirable (Gelfand and Ghosh 1998). The meanIC metric consists of 2 measures (one for precision and one for accuracy) constructed from the posterior predicted response variables for all (i,o,t) at all MCMC iterations. WAIC is invariant to parametrization and is asymptotically equal to Bayesian cross-validation (Watanabe 2010). WAIC = -2 * (lppd - p_waic_2). Where lppd is the log pointwise predictive density and p_waic_2 is the estimated effective number of parameters based on the variance estimator from Vehtari et al. 2016. (p_waic_1 is the mean estimator).

Value

diagnostics returns a list containing the diagnostics requested and possibly the deviances corresponding to all kept MCMC iterations.

References

Yifan Cheng

Berchuck, S. I., Janko, M., Medeiros, F. A., Pan, W., & Mukherjee, S. (2021). Bayesian Non-Parametric Factor Analysis for Longitudinal Spatial Surfaces. Bayesian Analysis, 17(2), 1–30.

Gelfand, A. E., & Ghosh, S. K. (1998). Model choice: a minimum posterior predictive loss approach. Biometrika, 1-11.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 64(4), 583-639.

Vehtari, A., Gelman, A., & Gabry, J. (2016). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. Statistics and Computing, 1-20.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. Journal of Machine Learning Research, 11(Dec), 3571-3594.

---

is.FixedLbfa *is.FixedLbfa*

Description

is.FixedLbfa is a general test of an object being interpretable as a FixedLbfa object.

Usage

```
is.FixedLbfa(x)
```

Arguments

x	object to be tested.

Details

The FixedLbfa class is defined as the regression object that results from the bfaFixedL regression function.

Value

is.FixedLbfa returns a logical, depending on whether the object is of class FixedLbfa.

Examples

```
###Load pre-computed results
data(reg.FixedLbfa)

###Test function
is.FixedLbfa(reg.FixedLbfa)
```

is.FixedLbfaVAR1 *is.FixedLbfaVAR1*

Description

is.FixedLbfaVAR1 is a general test of an object being interpretable as a FixedLbfaVAR1 object.

Usage

```
is.FixedLbfaVAR1(x)
```

Arguments

x	object to be tested.

Details

The FixedLbfaVAR1 class is defined as the regression object that results from the VAR1bfaFixedL regression function.

Value

is.FixedLbfaVAR1 returns a logical, depending on whether the object is of class FixedLbfaVAR1.

Examples

```
###Load pre-computed results
data(reg.FixedLbfaVAR1)

###Test function
is.FixedLbfaVAR1(reg.FixedLbfaVAR1)
```

is.VAR1varyLjBFA *is.VAR1varyLjBFA*

Description

is.VAR1varyLjBFA is a general test of an object being interpretable as a VAR1varyLjBFA object.

Usage

```
is.VAR1varyLjBFA(x)
```

Arguments

x	object to be tested.

Details

The VAR1varyLjBFA class is defined as the regression object that results from the VAR1bfaVaryingLjs regression function.

Value

is.VAR1varyLjBFA returns a logical, depending on whether the object is of class VAR1varyLjBFA.

Examples

```
###Load pre-computed results
data(reg.VAR1varyLjBFA)

###Test function
is.varyLjBFA(reg.VAR1varyLjBFA)
```

---

is.varyLjBFA *is.varyLjBFA*

Description

is.varyLjBFA is a general test of an object being interpretable as a varyLjBFA object.

Usage

```
is.varyLjBFA(x)
```

Arguments

x	object to be tested.

Details

The varyLjBFA class is defined as the regression object that results from the bfaVaryingLjs regression function.

Value

is.varyLjBFA returns a logical, depending on whether the object is of class varyLjBFA.

Examples

```
###Load pre-computed results
data(reg.varyLjBFA)

###Test function
is.varyLjBFA(reg.varyLjBFA)
```

predictNewLocFixedL *predictNewLocFixedL*

Description

Predicts observations at new location point(s) from a FixedLbfa or FixedLbfaVAR1 model.

Usage

```
predictNewLocFixedL(
  object,
  NNewLoc,
  distOrigNew = NULL,
  distNewNew = NULL,
  NewX = NULL,
  NewTrials = NULL,
  Verbose = TRUE,
  seed = 27
)
```

Arguments

object	A FixedLbfa model object for which predictions are desired from.
NNewLoc	A positive integer indicating the number of new spatial locations for prediction.
distOrigNew	An M x NNewLoc dimensional distance matrix for distances between the original and new spatial points. If include.space = FALSE, then we don't need to specify distOrigNew and can leave it as its default value NULL. If include.space = TRUE, then this matrix must be specified no matter the value of the input spatApprox. Since spatial.structure must be "continuous" to enable predictions at new spatial locations, the matrix should be a continuous distance matrix (e.g., Euclidean).
distNewNew	An NNewLoc x NNewLoc dimensional distance matrix for the new spatial points. If include.space = FALSE or spatApprox = TRUE, then we don't need to specify distNewNew and can leave it as its default value NULL. When include.space = TRUE, the matrix should be a continuous distance matrix (e.g., Euclidean), since spatial.structure must be "continuous" to enable predictions at new spatial locations.
NewX	A matrix including covariates at times 1:T for the new location points. NewX must have dimension (NNewLoc x O x T) x P, where NNewLoc is the number of new location points being predicted. The default sets NewX to NULL and assumes that the covariates for all new locations are the same as the ones corresponding to the last reference location point.
NewTrials	An array indicating the trials for categorical predictions. The array must have dimension T x C x NNewLoc and contain only non-negative integers. The default sets NewTrials to NULL and assumes that the trials for all predictions are the same as the ones corresponding to the final reference time point.
Verbose	A boolean logical indicating whether progress should be output.
seed	An integer value used to set the seed for the random number generator.

Details

predictNewLocFixedL predicts vectors at new spatial location(s). The function always returns the predicted factor loadings matrices Lambda and outcomes (Y). When clustering = TRUE, the function also returns the posterior predicted latent spatial vectors alpha for the new spatial location(s), which (when include.space = TRUE) are obtained via Bayesian kriging when spatApprox = FALSE and via nearest-neighbor kriging when spatApprox = TRUE. When clustering = TRUE and storeWeights = TRUE, the function also returns the posterior predicted weights for the new location(s) we want to predict.

Value

predictNewLocFixedL returns a list containing the following objects.

Alpha A matrix of dimension NKeep x (K x (L - 1) x NNewLoc x O) containing the posterior predicted latent spatial vectors alpha for the new location(s) we want to predict. NULL when clustering = FALSE when fitting the main function. For each kept row (MCMC iteration) in Alpha, the corresponding entries are ordered first by observation type, then spatially, then by clustering group, and finally by factor.

Weights A matrix of dimension NKeep x (K x L x NNewLoc x O) containing the posterior predicted weights for the new location(s) we want to predict. This output can be used for clustering the new location point(s). NULL when clustering = FALSE or storeWeights = FALSE when fitting the main function. For each kept row (MCMC iteration) in Weights, the corresponding entries are ordered first by observation type, then spatially, then by clustering group, and finally by factor.

Lambda A matrix of dimension NKeep x (NNewLoc x O x K) containing the posterior predicted factor loadings matrices, where O is the number of observation types and K is the number of latent factors. For each kept MCMC iteration, the corresponding predicted entries for the factor loadings matrix are ordered first by observation type, then spatially, and finally by factor. (the first (NNewLoc x O) entries correspond to factor 1, the next (NNewLoc x O) entries correspond to factor 2 and so on; the first O rows correspond to the first new location point for prediction, the next O rows correspond to the second new location point for prediction and so on)

Y A list containing NNewLoc posterior predictive distribution matrices. Each matrix is of dimension NKeep x (T x O), where T is the number of time points and O is the number of observation types. For each kept MCMC iteration, the values are ordered first temporally and then by observation type (the first T correspond to the first observation type, the next T correspond to the second observation type and so on). Each matrix is obtained through Bayesian krigging.

alphaKrigTime Time (in milliseconds) it took to obtain Alpha. NULL when clustering = FALSE when fitting the main function.

weightsXiLambdaKrigTime Time (in milliseconds) it took to obtain Weights, Xi, and Lambda from Alpha and posterior samples of theta. NULL when clustering = FALSE when fitting the main function.

lambdaKrigTime Time (in milliseconds) it took to obtain Lambda. NULL when clustering = TRUE when fitting the main function.

References

Yifan Cheng

predictNewLocVaryLj *predictNewLocVaryLj*

Description

Predicts observations at new location point(s) from a varyLjBFA or VAR1varyLjBFA model.

Usage

```
predictNewLocVaryLj(
  object,
  NNewLoc,
  distOrigNew = NULL,
  distNewNew = NULL,
  NewX = NULL,
  NewTrials = NULL,
  Verbose = TRUE,
  seed = 27
)
```

Arguments

object	A varyLjBFA model object for which predictions are desired from.
NNewLoc	A positive integer indicating the number of new spatial locations for prediction.
distOrigNew	An M x NNewLoc dimensional distance matrix for distances between the original and new spatial points. If include.space = FALSE, then we don't need to specify distOrigNew and can leave it as its default value NULL. If include.space = TRUE, then this matrix must be specified no matter the value of the input spatApprox. Since spatial.structure must be "continuous" to enable predictions at new spatial locations, the matrix should be a continuous distance matrix (e.g., Euclidean).
distNewNew	An NNewLoc x NNewLoc dimensional distance matrix for the new spatial points. If include.space = FALSE or spatApprox = TRUE, then we don't need to specify distNewNew and can leave it as its default value NULL. When include.space = TRUE, the matrix should be a continuous distance matrix (e.g., Euclidean), since spatial.structure must be "continuous" to enable predictions at new spatial locations.
NewX	A matrix including covariates at times 1:T for the new location points. NewX must have dimension (NNewLoc x O x T) x P, where NNewLoc is the number of new location points being predicted. The default sets NewX to NULL and assumes that the covariates for all new locations are the same as the ones corresponding to the last reference location point.
NewTrials	An array indicating the trials for categorical predictions. The array must have dimension T x C x NNewLoc and contain only non-negative integers. The default sets NewTrials to NULL and assumes that the trials for all predictions are the same as the ones corresponding to the final reference time point.
Verbose	A boolean logical indicating whether progress should be output.
seed	An integer value used to set the seed for the random number generator.

Details

predictNewLocVaryLj predicts vectors at new spatial location(s). The function returns the predicted factor loadings matrices Lambda and outcomes (Y). The function also returns the posterior predicted latent spatial vectors alpha and the corresponding weights for the new spatial location(s), which (when include.space = TRUE) are obtained via Bayesian kriging when spatApprox = FALSE and via nearest-neighbor kriging when spatApprox = TRUE.

Value

predictNewLocVaryLj returns a list containing the following objects.

Alpha A list of posterior predicted values for alpha. The list contains K x NKeep elements (ordered by column), each of which is a matrix of dimension (NNewLoc x O) x (Lj - 1) for Lj at that MCMC iteration. If Lj = 1 for a certain j at a particular MCMC iteration, then that corresponding matrix in Alpha is of dimension (NNewLoc x O) x 1 with entries all set to +Inf. In each element matrix for each clustering group, the corresponding entries are ordered first by observation type and then spatially. (the first O correspond to the first new location point for prediction, the next O correspond to the second new location point for prediction and so on)

Weights A list of posterior predicted values for weights. The list contains K x NKeep elements (ordered by column), each of which is a matrix of dimension (NNewLoc x O) x Lj for Lj at that MCMC iteration. In each element matrix for each clustering group, the corresponding entries are ordered first by observation type and then spatially. (the first O correspond to the first new location point for prediction, the next O correspond to the second new location point for prediction and so on)

Lambda A matrix of dimension NKeep x (NNewLoc x O x K) containing the posterior predicted factor loadings matrices, where O is the number of observation types and K is the number of latent factors. For each kept MCMC iteration, the corresponding predicted entries for the factor loadings matrix are ordered first by observation type, then spatially, and finally by factor. (the first (NNewLoc x O) entries correspond to factor 1, the next (NNewLoc x O) entries correspond to factor 2 and so on; the first O rows correspond to the first new location point for prediction, the next O rows correspond to the second new location point for prediction and so on)

Y A list containing NNewLoc posterior predictive distribution matrices. Each matrix is of dimension NKeep x (T x O), where T is the number of time points and O is the number of observation types. For each kept MCMC iteration, the values are ordered first temporally and then by observation type (the first T correspond to the first observation type, the next T correspond to the second observation type and so on). Each matrix is obtained through Bayesian krigging.

alphaKrigTime Time (in milliseconds) it took to obtain Alpha.,

weightsXiLambdaKrigTime Time (in milliseconds) it took to obtain Weights, Xi, and Lambda from Alpha and posterior samples of theta.

References

Yifan Cheng

predictNewTime *predictNewTime*

Description

Predicts future observation(s) from a FixedLbfa or varyLjBFA model.

Usage

```
predictNewTime(
  object,
  NewTimes,
  NewX = NULL,
  NewTrials = NULL,
  Verbose = TRUE,
  seed = 54,
  ...
)
```

Arguments

object	A FixedLbfa or varyLjBFA model object for which predictions are desired from.
NewTimes	A numeric vector including desired time point(s) for prediction. If include.time = equalTimeDist = TRUE when getting this function's input object from the main function bfa_sp and the pre-normalized time distance does not equal 1, then we should standardize the new time points accordingly. NewTimes should be the standardized desired time point(s) for prediction.
NewX	A matrix including covariates at times NewTimes for prediction. NewX must have dimension (M x O x NNewVisits) x P. Where NNewVisits is the number of temporal locations being predicted. The default sets NewX to NULL, which assumes that the covariates for all predictions are the same as the final time point.
NewTrials	An array indicating the trials for categorical predictions. The array must have dimension M x C x NNewVisits and contain only non-negative integers. The default sets NewTrials to NULL, which assumes that trials for all predictions are the same as the final time point.
Verbose	A boolean logical indicating whether progress should be output.
seed	An integer value used to set the seed for the random number generator (default = 54).
...	other arguments.

Details

predictNewTime uses Bayesian krigging to predict vectors at future time points. The function returns the krigged factors (Eta) and also the observed outcomes (Y).

Value

predictNewTime returns a list containing the following objects.

Eta A list containing NNewVisits matrices, one for each new time prediction. Each matrix is dimension NKeep x K, where K is the number of latent factors. Each matrix contains posterior samples obtained by Bayesian krigging.

Y A list containing NNewVisits posterior predictive distribution matrices. Each matrix is of dimension NKeep x (M * O), where M is the number of spatial locations and O the number of observation types. Each matrix is obtained through Bayesian krigging.

References

Yifan Cheng

Berchuck, S. I., Janko, M., Medeiros, F. A., Pan, W., & Mukherjee, S. (2021). Bayesian Non-Parametric Factor Analysis for Longitudinal Spatial Surfaces. Bayesian Analysis, 17(2), 1–30.

---

spatempBFA *spatempBFA*

Description

spatempBFA

VAR1bfaFixedL *Spatial factor analysis using a Bayesian hierarchical model.*

Description

VAR1bfaFixedL is a Markov chain Monte Carlo (MCMC) sampler for a Bayesian spatial factor analysis model. The spatial component is introduced using a Probit stick-breaking process prior on the factor loadings. The model is implemented using a Bayesian hierarchical framework.

Usage

```
VAR1bfaFixedL(
  formula,
  data,
  dist,
  Nu,
  K,
  L = 20,
  family = "normal",
  spatial.structure = "continuous",
  starting = NULL,
  hypers = NULL,
  tuning = NULL,
  mcmc = NULL,
  seed = 27,
```

```
  gamma.shrinkage = TRUE,
  include.space = TRUE,
  clustering = TRUE,
  spatApprox = TRUE,
  alphaMethod = "block",
  h = 15,
  storeSpatPredPara = TRUE,
  storeWeights = TRUE,
  alphasWeightsToFiles = FALSE
)
```

Arguments

formula	A formula object, corresponding to the spatial factor analysis model. The response must be on the left of a ~ operator, and the terms on the right must indicate the covariates to be included in the fixed effects. If no covariates are desired a zero should be used, ~ 0.
data	A required data.frame containing the variables (Y, additional x coviariate(s) if there is/are, and a trials variable if there is/are observation type(s) from family "binomial") in the model. The data frame must contain M x O x Nu rows. Here, M represents the number of spatial locations, O the number of different observation types and Nu the number of temporal visits. The observations must be first ordered spatially, second by observation type and then temporally. This means that the first M x O observations come from the first time point and the first M observations come the first spatial observation type. If there is/are observation type(s) from family "binomial", then trials contains the numbers of trials as positive integers for each of the binomial observations. Entries in trials corresponding to non-binomial data can be specified as any arbitrary positive integer. The function will change these values to 1.
dist	A M x M dimensional distance matrix. For a discrete spatial process the matrix contains binary adjacencies that dictate the spatial neighborhood structure and for continuous spatial processes the matrix should be a continuous distance matrix (e.g., Euclidean).
Nu	A positive integer representing the number of evenly dispersed time points corresponding to our observed data.
K	A scalar that indicates the dimension (i.e., quantity) of latent factors.
L	A positive integer indicating the number of latent clusters for each column of the factor loadings matrix.
family	A character string or a vector of length O (if O > 1) of character strings indicating the distribution(s) of the observed data. Options for each observation type include: "normal", "probit", "tobit", and "binomial". If O > 1 and family is of length 1, then all of the O observation types are from the same family (distribution). Any combination of likelihoods can be used.
spatial.structure	Character string indicating the type of spatial process. Options include: "continuous" (i.e., Gaussian process with exponential kernel) and "discrete" (i.e., proper CAR).
starting	Either NULL or a list containing starting values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the starting values may be specified.

When NULL is chosen then default starting values are automatically generated. Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon, or A containing appropriate objects. Beta (or Delta) must either be a P (or K) dimensional vector or a scalar (the scalar populates the entire vector). Sigma2 must be either a M x (O - C) matrix or a scalar. Kappa must be a O x O dimensional matrix, Rho a scalar, and Upsilon a K x K matrix.

hypers

Either NULL or a list containing hyperparameter values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the hyperparameter values may be specified.

When NULL is chosen then default hyperparameter values are automatically generated. These default hyperparameters are described in detail in (Berchuck et al.). Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon, or A containing further hyperparameter information. These objects are themselves lists and may be constructed as follows.

Beta is a list with two objects, MuBeta and SigmaBeta. These values represent the prior mean and variance parameters for the multivariate normal prior.

Delta is a list with two objects, A1 and A2. These values represent the prior shape parameters for the multiplicative Gamma shrinkage prior.

Sigma2 is a list with two objects, A and B. These values represent the shape and scale for the variance parameters.

Kappa is a list with two objects, SmallUpsilon and BigTheta. SmallUpsilon represents the degrees of freedom parameter for the inverse-Wishart hyperprior and must be a real number scalar, while BigTheta represents the scale matrix and must be a O x O dimensional positive definite matrix.

Rho is a list with two objects, ARho and BRho. ARho represents the lower bound for the uniform hyperprior, while BRho represents the upper bound. The bounds must be specified carefully. This is only specified for continuous spatial processes.

Upsilon is a list with two objects, Zeta and Omega. Zeta represents the degrees of freedom parameter for the inverse-Wishart hyperprior and must be a real number scalar, while Omega represents the scale matrix and must be a K x K dimensional positive definite matrix.

A is a list with two objects, V and Mvec. V is a K x K dimensional positive definite matrix, and Mvec is a scale vector of length K^2.

tuning

Either NULL or a list containing tuning values to be specified for the MCMC Metropolis steps. If NULL is not chosen then all of the tuning values must be specified.

When NULL is chosen then default tuning values are automatically generated to 1. Otherwise a list must be provided with name Rho. Each of these entries must be scalars containing tuning variances for their corresponding Metropolis updates.

mcmc

Either NULL or a list containing input values to be used for implementing the MCMC sampler. If NULL is not chosen then all of the MCMC input values must be specified.

NBurn: The number of sampler scans included in the burn-in phase. (default = 10,000)

NSims: The number of post-burn-in scans for which to perform the sampler. (default = 10,000)

NThin: Value such that during the post-burn-in phase, only every NThin-th scan is recorded for use in posterior inference (For return values we define, NKeep = NSims / NThin (default = 1).

NPilot: The number of times during the burn-in phase that pilot adaptation is performed (default = 20)

seed An integer value used to set the seed for the random number generator

gamma.shrinkage
A logical indicating whether a gamma shrinkage process prior is used for the variances of the factor loadings columns. If FALSE, the hyperparameters (A1 and A2) indicate the shape and rate for a gamma prior on the precisions. Default is TRUE. It can only be TRUE when clustering = TRUE.

include.space A logical indicating whether a spatial process should be included. Default is TRUE, however if FALSE the spatial correlation matrix F(\rho) is fixed as the M x M identity matrix. This specification overrides the inputs spatial.structure, spatApprox, and alphaMethod.

clustering A logical indicating whether the Bayesian non-parametric process should be used, default is TRUE. If FALSE is specified each column is instead modeled with an independent spatial process.

spatApprox A logical indicating whether spatial nearest neighbor kriging is used for the latent location-specific spatial parameter vectors \alpha_{jl_j}'s to accelerate computation. Default is TRUE. When TRUE, spatial.structure must be "continuous". If FALSE, the exact M x M neighborhood structure matrix F(\rho) is adopted throughout.

alphaMethod Character string indicating which approach is used to update the \alpha_{jl_j}'s in the Gibbs Sampler. Options include: "block" and "sequential". This argument is only used when include.space = TRUE and spatApprox = TRUE.

h a positive integer much smaller than M, the number of location points, indicating an upper bound of the number of nearest neighbors for each location point.

storeSpatPredPara
a logical indicating whether we want to store the posterior samples for Xi, Delta, Tau, Alpha, Theta when clustering = TRUE. Has to be TRUE if we want to perform predictions at new spatial location(s). Can be FALSE to save storage space if spatial prediction is not required.

storeWeights a logical indicating whether we want to store the posterior Weights estimates when clustering = TRUE. Has to be TRUE if we want to perform temporal trends clustering from the output object. Can be FALSE to save storage space if otherwise.

alphasWeightsToFiles
a logical indicating whether we want to store the posterior alpha and weights estimates to files (one for each kept MCMC iteration) instead of including them in the main output list. Can only be TRUE if clustering = TRUE and at least one of storeSpatPredPara and storeWeights is TRUE. When that is the case, specifying alphasWeightsToFiles = TRUE would be immensely useful if we don't have enough space to output a model fit object with all kept posterior parameter estimates. Default is FALSE.

Value

VAR1bfaFixedL returns a list containing the following objects (some may be NULL)

lambda NKeep x (M x O x K) matrix of posterior samples for the factor loadings matrix Lambda. The labels for each column are Lambda_O_M_K.

eta NKeep x (Nu x K) matrix of posterior samples for the latent factors eta. The labels for each column are Eta_Nu_K.

beta NKeep x P matrix of posterior samples for beta.

sigma2 NKeep x (M * (O - C)) matrix of posterior samples for the variances sigma2. The labels for each column are Sigma2_O_M.

kappa NKeep x ((O * (O + 1)) / 2) matrix of posterior samples for kappa. The columns have names that describe the samples within them. The row is listed first, e.g., Kappa3_2 refers to the entry in row 3, column 2.

delta NKeep x K matrix of posterior samples for delta.

tau NKeep x K matrix of posterior samples for tau.

upsilon NKeep x ((K * (K + 1)) / 2) matrix of posterior samples for Upsilon. The columns have names that describe the samples within them. The row is listed first, e.g., Upsilon3_2 refers to the entry in row 3, column 2.

A NKeep x (K * K) matrix of posterior samples for A.

xi NKeep x (M x O x K) matrix of posterior samples for factor loadings cluster labels xi. The labels for each column are Xi_O_M_K.

rho NKeep x 1 matrix of posterior samples for rho.

theta NKeep x (L x K) matrix of posterior samples for theta.

alpha NKeep x (M x O x K x (L - 1) matrix of posterior samples for alpha. For each kept row (MCMC iteration) in alpha, the corresponding entries are ordered first by observation type, then spatially, then by clustering group, and finally by factor.

weights NKeep x (M x O x K x L) matrix of posterior samples for weights. For each kept row (MCMC iteration) in weights, the corresponding entries are ordered first spatially, then by observation type, then by clustering group, and finally by factor.

metropolis 2 (or 1) x 3 matrix of metropolis acceptance rates, updated tuners, and original tuners that result from the pilot adaptation.

datobj A list of data objects that are used in future bfaFixedL functions and should be ignored by the user.

dataug A list of data augmentation objects that are used in future bfaFixedL functions and should be ignored by the user.

runtime A character string giving the runtime of the MCMC sampler.

VAR1bfaVaryingLjs *Spatial factor analysis using a Bayesian hierarchical model.*

Description

VAR1bfaVaryingLjs is a Markov chain Monte Carlo (MCMC) sampler for a Bayesian spatial factor analysis model. The spatial component is introduced using a Probit stick-breaking process prior on the factor loadings. The model is implemented using a Bayesian hierarchical framework.

Usage

```
VAR1bfaVaryingLjs(
  formula,
  data,
  dist,
  Nu,
```

```
  K,
  LjVec,
  family = "normal",
  spatial.structure = "continuous",
  starting = NULL,
  hypers = NULL,
  tuning = NULL,
  mcmc = NULL,
  seed = 27,
  gamma.shrinkage = TRUE,
  include.space = TRUE,
  spatApprox = TRUE,
  alphaSequen = FALSE,
  h = 15,
  storeSpatPredPara = TRUE,
  storeWeights = TRUE
)
```

Arguments

formula	A formula object, corresponding to the spatial factor analysis model. The response must be on the left of a ~ operator, and the terms on the right must indicate the covariates to be included in the fixed effects. If no covariates are desired a zero should be used, ~ 0.
data	A required data.frame containing the variables (Y, additional x coviariate(s) if there is/are, and a trials variable if there is/are observation type(s) from family "binomial") in the model. The data frame must contain M x O x Nu rows. Here, M represents the number of spatial locations, O the number of different observation types and Nu the number of temporal visits. The observations must be first ordered spatially, second by observation type and then temporally. This means that the first M x O observations come from the first time point and the first M observations come the first spatial observation type. If there is/are observation type(s) from family "binomial", then trials contains the numbers of trials as positive integers for each of the binomial observations. Entries in trials corresponding to non-binomial data can be specified as any arbitrary positive integer. The function will change these values to 1.
dist	A M x M dimensional distance matrix. For a discrete spatial process the matrix contains binary adjacencies that dictate the spatial neighborhood structure and for continuous spatial processes the matrix should be a continuous distance matrix (e.g., Euclidean).
Nu	A positive integer representing the number of evenly dispersed time points corresponding to our observed data.
K	A scalar that indicates the dimension (i.e., quantity) of latent factors.
LjVec	A vector of length K consisting of positive integers indicating the starting numbers of latent clusters for the K columns of the factor loadings matrix.
family	A character string or a vector of length O (if O > 1) of character strings indicating the distribution(s) of the observed data. Options for each observation type include: "normal", "probit", "tobit", and "binomial". If O > 1 and family is of length 1, then all of the O observation types are from the same family (distribution). Any combination of likelihoods can be used.

spatial.structure
: Character string indicating the type of spatial process. Options include: "continuous" (i.e., Gaussian process with exponential kernel) and "discrete" (i.e., proper CAR).

starting
: Either NULL or a list containing starting values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the starting values may be specified.

When NULL is chosen then default starting values are automatically generated. Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon, or A containing appropriate objects. Beta (or Delta) must either be a P (or K) dimensional vector or a scalar (the scalar populates the entire vector). Sigma2 must be either a M x (O - C) matrix or a scalar. Kappa must be a O x O dimensional matrix, Rho a scalar, Upsilon a K x K matrix, and Psi a scalar.

hypers
: Either NULL or a list containing hyperparameter values to be specified for the MCMC sampler. If NULL is not chosen then none, some or all of the hyperparameter values may be specified.

When NULL is chosen then default hyperparameter values are automatically generated. These default hyperparameters are described in detail in (Berchuck et al.). Otherwise a list must be provided with names Beta, Delta, Sigma2, Kappa, Rho, Upsilon, or A containing further hyperparameter information. These objects are themselves lists and may be constructed as follows.

Beta is a list with two objects, MuBeta and SigmaBeta. These values represent the prior mean and variance parameters for the multivariate normal prior.

Delta is a list with two objects, A1 and A2. These values represent the prior shape parameters for the multiplicative Gamma shrinkage prior.

Sigma2 is a list with two objects, A and B. These values represent the shape and scale for the variance parameters.

Kappa is a list with two objects, SmallUpsilon and BigTheta. SmallUpsilon represents the degrees of freedom parameter for the inverse-Wishart hyperprior and must be a real number scalar, while BigTheta represents the scale matrix and must be a O x O dimensional positive definite matrix.

Rho is a list with two objects, ARho and BRho. ARho represents the lower bound for the uniform hyperprior, while BRho represents the upper bound. The bounds must be specified carefully. This is only specified for continuous spatial processes.

Upsilon is a list with two objects, Zeta and Omega. Zeta represents the degrees of freedom parameter for the inverse-Wishart hyperprior and must be a real number scalar, while Omega represents the scale matrix and must be a K x K dimensional positive definite matrix.

A is a list with two objects, V and Mvec. V is a K x K dimensional positive definite matrix, and Mvec is a scale vector of length K^2.

tuning
: Either NULL or a list containing tuning values to be specified for the MCMC Metropolis steps. If NULL is not chosen then all of the tuning values must be specified.

When NULL is chosen then default tuning values are automatically generated to 1. Otherwise a list must be provided with name Rho. Each of these entries must be scalars containing tuning variances for their corresponding Metropolis updates.

mcmc
: Either NULL or a list containing input values to be used for implementing the MCMC sampler. If NULL is not chosen then all of the MCMC input values must be specified.

NBurn: The number of sampler scans included in the burn-in phase. (default = 10,000)

NSims: The number of post-burn-in scans for which to perform the sampler. (default = 10,000)

NThin: Value such that during the post-burn-in phase, only every NThin-th scan is recorded for use in posterior inference (For return values we define, NKeep = NSims / NThin (default = 1).

NPilot: The number of times during the burn-in phase that pilot adaptation is performed (default = 20)

seed An integer value used to set the seed for the random number generator

gamma.shrinkage
A logical indicating whether a gamma shrinkage process prior is used for the variances of the factor loadings columns. If FALSE, the hyperparameters (A1 and A2) indicate the shape and rate for a gamma prior on the precisions. Default is TRUE.

include.space A logical indicating whether a spatial process should be included. Default is TRUE, however if FALSE the spatial correlation matrix F(\rho) is fixed as the M x M identity matrix. This specification overrides the inputs spatial.structure, spatApprox, and alphaMethod.

spatApprox A logical indicating whether spatial nearest neighbor kriging is used for the latent location-specific spatial parameter vectors \alpha_{jl_j}'s to accelerate computation. Default is TRUE. When TRUE, spatial.structure must be "continuous". If FALSE, the exact M x M neighborhood structure matrix F(\rho) is adopted throughout.

alphaSequen A logical indicating whether we want to sequentially update the alpha's. This argument is only used when include.space = TRUE and spatApprox = TRUE.

h a positive integer much smaller than M, the number of location points, indicating an upper bound of the number of nearest neighbors for each location point.

storeSpatPredPara
a logical indicating whether we want to store the posterior samples for Alpha, Theta. Has to be TRUE if we want to perform predictions at new spatial location(s). Can be FALSE to save storage space if spatial prediction is not required.

storeWeights a logical indicating whether we want to store the posterior Weights estimates. Has to be TRUE if we want to perform temporal trends clustering from the output object. Can be FALSE to save storage space if otherwise.

Value

VAR1bfaVaryingLjs returns a list containing the following objects (some may be NULL)

lambda NKeep x (M x O x K) matrix of posterior samples for the factor loadings matrix Lambda. The labels for each column are Lambda_O_M_K.

eta NKeep x (Nu x K) matrix of posterior samples for the latent factors eta. The labels for each column are Eta_Nu_K.

beta NKeep x P matrix of posterior samples for beta.

sigma2 NKeep x (M * (O - C)) matrix of posterior samples for the variances sigma2. The labels for each column are Sigma2_O_M.

kappa NKeep x ((O * (O + 1)) / 2) matrix of posterior samples for kappa. The columns have names that describe the samples within them. The row is listed first, e.g., Kappa3_2 refers to the entry in row 3, column 2.

`delta` `NKeep x K matrix` of posterior samples for `delta`.

`tau` `NKeep x K matrix` of posterior samples for `tau`.

`upsilon` `NKeep x ((K * (K + 1)) / 2) matrix` of posterior samples for `Upsilon`. The columns have names that describe the samples within them. The row is listed first, e.g., `Upsilon3_2` refers to the entry in row 3, column 2.

`A` `NKeep x (K * K) matrix` of posterior samples for `A`.

`xi` `NKeep x (M x O x K) matrix` of posterior samples for factor loadings cluster labels `xi`. The labels for each column are Xi_O_M_K.

`rho` `NKeep x 1 matrix` of posterior samples for `rho`.

`ljvec` `NKeep x K matrix` of posterior samples for `Lj`'s.

`theta` A `list` of posterior samples for `theta`. The list contains `K x NKeep` elements (ordered by column), each of which is a vector of length `Lj` at that MCMC iteration.

`alpha` A `list` of posterior samples for `alpha`. The list contains `K x NKeep` elements (ordered by column, i.e., the first K components of the list correspond to the first kept iteration and so on), each of which is a `matrix` of dimension `(Lj - 1) x (M x O)` for `Lj` at that MCMC iteration. If `Lj = 1` for a certain `j` at a particular MCMC iteration, then that corresponding `matrix` in `alpha` is of dimension `1 x (M x O)` with entries all set to +Inf. In each element matrix for each clustering group, the corresponding entries are ordered first by observation type, then spatially. (the first O correspond to the first location point, the next O correspond to the second location point and so on)

`weights` A `list` of posterior samples for `weights`. The list contains `K x NKeep` elements (ordered by column), each of which is a `matrix` of dimension `Lj x (M x O)` for `Lj` at that MCMC iteration. In each element matrix for each clustering group, the corresponding entries are ordered first spatially, then by observation type. (the first M correspond to the first observation type, the next M correspond to the second observation type and so on)

`metropolis` `2 (or 1) x 3 matrix` of metropolis acceptance rates, updated tuners, and original tuners that result from the pilot adaptation.

`datobj` A `list` of data objects that are used in future `bfaVaryingLjs` functions and should be ignored by the user.

`dataug` A `list` of data augmentation objects that are used in future `bfaVaryingLjs` functions and should be ignored by the user.

`runtime` A `character` string giving the runtime of the MCMC sampler.

`VAR1predictNewTime` *predictNewTime*

Description

Predicts future observation(s) from a `FixedLbfaVAR1` or `VAR1varyLjBFA` model.

Usage

```
VAR1predictNewTime(
  object,
  NNewTime,
  NewX = NULL,
  NewTrials = NULL,
```

```
  Verbose = TRUE,
  seed = 54,
  ...
)
```

Arguments

object	A FixedLbfaVAR1 or VAR1varyLjBFA model object for which predictions are desired from.
NNewTime	An integer indicating the desired number of consecutive future time point(s) after T for prediction.
NewX	A matrix including covariates at times NewTimes for prediction. NewX must have dimension (M x O x NNewTime) x P. Where NNewTime is the number of temporal locations being predicted. The default sets NewX to NULL, which assumes that the covariates for all predictions are the same as the final time point.
NewTrials	An array indicating the trials for categorical predictions. The array must have dimension M x C x NNewTime and contain only non-negative integers. The default sets NewTrials to NULL, which assumes that trials for all predictions are the same as the final time point.
Verbose	A boolean logical indicating whether progress should be output.
seed	An integer value used to set the seed for the random number generator (default = 54).
...	other arguments.

Details

predictNewTime uses Bayesian krigging to predict vectors at future time points. The function returns the krigged factors (Eta) and also the observed outcomes (Y).

Value

predictNewTime returns a list containing the following objects.

Eta A list containing NNewTime matrices, one for each new time prediction. Each matrix is dimension NKeep x K, where K is the number of latent factors. Each matrix contains posterior samples obtained by Bayesian krigging.

Y A list containing NNewTime posterior predictive distribution matrices. Each matrix is of dimension NKeep x (M * O), where M is the number of spatial locations and O the number of observation types. Each matrix is obtained through Bayesian krigging.

References

Yifan Cheng

Index