Package: BayesMixSurv (via r-universe)

August 20, 2024

Type Package

Title Bayesian Mixture Survival Models using Additive Mixture-of-Weibull Hazards, with Lasso Shrinkage and Stratification

Version 0.9.1

Date 2016-09-08

Author Alireza S. Mahani, Mansour T.A. Sharabiani

Maintainer Alireza S. Mahani <alireza.s.mahani@gmail.com>

Description Bayesian Mixture Survival Models using Additive Mixture-of-Weibull Hazards, with Lasso Shrinkage and Stratification. As a Bayesian dynamic survival model, it relaxes the proportional-hazard assumption. Lasso shrinkage controls overfitting, given the increase in the number of free parameters in the model due to presence of two Weibull components in the hazard function.

License GPL (>= 2)

Depends survival

NeedsCompilation no

Date/Publication 2016-09-08 10:24:27

Repository https://asmahani.r-universe.dev

RemoteUrl https://github.com/cran/BayesMixSurv

RemoteRef HEAD

RemoteSha 7fd49fa6d9d57240fd8dff33b53ab377f94bcf67

Contents

bayesmixsurv *Dynamic Bayesian survival model - with stratification and Lasso shrinkage - for right-censored data using two-component additive mixture-of-Weibull hazards.*

Description

Bayesian survival model for right-censored data, using a sum of two hazard functions, each having a power dependence on time, corresponding to a Weibull distribution on event density. (Note that event density function for the mixture model does NOT remain a Weibull distribution.) Each component has a different shape and scale parameter, with scale parameters each being the exponential of a linear function of covariates specified in `formula1` and `formula2`. Stratification is implemented using a common set of intercepts between the two components. Lasso shrinkage - using Laplace prior on coefficients (Park and Casella 2008) - allows for variable selection in the presence of low observation-to-variable ratio. The mixture model allows for time-dependent (and context-dependent) hazard ratios. Confidence intervals for coefficient estimation and prediction are generated using full Bayesian paradigm, i.e. by keeping all samples rather than summarizing them into mean and sd. Posterior distribution is estimated via MCMC sampling, using univariate slice sampler with stepout and shrinkage (Neal 2003).

Usage

```
bayesmixsurv(formula1, data, formula2=formula1, stratCol=NULL, weights, subset
  , na.action=na.fail, control=bayesmixsurv.control(), print.level=2)
bayesmixsurv.control(single=FALSE, alpha2.fixed=NULL, alpha.boundary=1.0, lambda1=1.0
  , lambda2=lambda1, iter=1000, burnin=round(iter/2), sd.thresh=1e-4, scalex=TRUE
  , nskip=round(iter/10))
## S3 method for class 'bayesmixsurv'
print(x, ...)
```

Arguments

`formula1`	Survival formula expressing the time/status variables as well as covariates used in the first component.
`data`	Data frame containing the covariates and response variable, as well as the stratification column.
`formula2`	Survival formula expressing the covariates used in the second component. No left-hand side is necessary since the response variable information is extracted from `formula1`. Defaults to `formula1`.
`stratCol`	Name of column in `data` used for stratification. Must be a factor or coerced into one. Default is no stratification (`stratCol=NULL`).
`weights`	Optional vector of case weights. *Not supported yet*
`subset`	Subset of the observations to be used in the fit. *Not supported yet*
`na.action`	Missing-data filter function. *Not supported yet (only na.fail behavior works)*

control	See bayesmixsurv.control for a description of the parameters inside the control list.
print.level	Controlling verbosity level.
single	If TRUE, a single-component model, equivalent to Bayesian Weibull survival regression, with Lasso shrinkage, is implemented. Default is FALSE, i.e. a two-component mixture-of-Weibull model.
alpha2.fixed	If provided, it specifies the shape parameter of the second component. Default is NULL, which allows the MCMC sampling to estimate both shape parameters.
alpha.boundary	When single=FALSE and alpha2.fixed=NULL, this parameter specifies an upper bound for the shape parameter of the first component, and a lower bound for the shape parameter of the second component. These boundary conditions are enforced in the univariate slice sampler function calls.
lambda1	Lasso Shrinkage parameter used in the Laplace prior on covariates used in the first component.
lambda2	Lasso Shrinkage parameter used in the Laplace prior on covariates used in the second component. Defaults to lambda1.
iter	Number of posterior MCMC samples to generate.
burnin	Number of initial MCMC samples to discard before calculating summary statistics.
sd.thresh	Threshold for standard deviation of a covariate (after possible centering/scaling). If below the threshold, the corresponding coefficient is removed from sampling, i.e. its value is clamped to zero.
scalex	If TRUE, each covariate vector is centered and scaled before model estimation. The scaling parameters are saved in return object, and used in subsequent calls to predict function. Users are strongly advised against turning this feature off, since the quality of Gibbs sampling MCMC is greatly enhanced by covariate centering and scaling.
nskip	Controlling how often to print progress report during MCMC run. For example, if nskip=10, progress will be reported after 10,20,30,... samples.
x	Object of class 'bayesmixsurv', usually the result of a call to bayesmixsurv.
...	Arguments to be passed to/from other methods.

Value

The function bayesmixsurv.control return a list with the same elements as its input parameters. The function bayesmixsurv returns object of class bayesmixsurv, with the following components:

call	The matched call
formula1	Same as input.
formula2	Same as input.
weights	Same as input. *Not supported yet*
subset	Same as input. *Not supported yet*
na.action	Same as input. *Not supported yet* (current behavior is na.fail)

control	Same as input.
X1	Model matrix used for component 1, after potential centering and scaling.
X2	Model matrix used for component 2, after potential centering and scaling.
y	Survival response variable (time and status) used in the model.
contrasts1	The contrasts used for component 1 (where relevant).
contrasts2	The contrasts used for component 2 (where relevant).
xlevels1	A record of the levels of the factors used in fitting for component 1 (where relevant).
xlevels2	A record of the levels of the factors used in fitting for component 2 (where relevant).
terms1	The terms object used for component 1.
terms2	The terms object used for component 2.
colnamesX1	Names of columns for X1, also names of scale coefficients for component 1.
colnamesX2	Names of columns for X1, also names of scale coefficients for component 2.
apply.scale.X1	Index of columns of X1 where scaling has been applied.
apply.scale.X2	Index of columns of X2 where scaling has been applied.
centerVec.X1	Vector of centering parameters for columns of X1 indicated by apply.scale.X1.
centerVec.X2	Vector of centering parameters for columns of X2 indicated by apply.scale.X2.
scaleVec.X1	Vector of scaling parameters for columns of X1 indicated by apply.scale.X1.
scaleVec.X2	Vector of scaling parameters for columns of X2 indicated by apply.scale.X2.
Xg	Model matrix associated with stratification (if any).
stratContrasts	The contrasts used for stratification model matrix, if any.
stratXlevels	A record of the levels of the factors used in stratification (if any)).
stratTerms	The terms object used for stratification.
colnamesXg	Names of columns for Xg.
idx1	Vector of indexes into X1 for which sampling occured. All columns of X1 whose standard deviation falls below sd.thresh are excluded from sampling and their corresponding coefficients are clamped to 0.
idx2	Vector of indexes into X2 for which sampling occured. All columns of X2 whose standard deviation falls below sd.thresh are excluded from sampling and their corresponding coefficients are clamped to 0.
median	List of median values, with elements including alpha1,alpha2 (shape parameter of components 1 and 2), beta1,beta2 (coefficients of scale parameter for components 1 and 2), gamma (stratification intercept adjustments, shared by 2 comoponents), and sigma.gamma (standard deviation of zero-mean Gaussian distribution that is the prior for gamma's).
max	Currently, a list with one element, loglike, containing the maximum sampled log-likelihood of the model.

smp — List of coefficient samples, with elements alpha1,alpha2 (shape parameters for components 1 and 2), beta1,beta2 (scale parameter coefficients for components 1 and 2), loglike (model log-likelihood), gamma (stratification intercept adjustments, shared by 2 comoponents), and sigma.gamma (standard deviation of zero-mean Gaussian distribution that is the prior for gamma's). Each parameter has iter samples. For vector parameters, first dimension is the number of samples (iter), while the second dimension is the length of the vector.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

References

Neal R.M. (2003). Slice Sampling. *Annals of Statistics*, **31**, 705-767.

Park T. and Casella G. (2008) The Bayesian Lasso. *Journal of the American Statistical Association*, **103**, 681-686.

Examples

```
# NOTE: to ensure convergence, typically more than 100 samples are needed
# fit the most general model, with two Weibull components and unspecified shape parameters
ret <- bayesmixsurv(Surv(time, status)~as.factor(trt)+age+as.factor(celltype)+prior, veteran
  , control=bayesmixsurv.control(iter=100))
# fix one of the two shape parameters
ret2 <- bayesmixsurv(Surv(time, status)~as.factor(trt)+age+as.factor(celltype)+prior, veteran
    , control=bayesmixsurv.control(iter=100, alpha2.fixed=1.0))
```

---

bayesmixsurv.crossval *Convenience functions for cross-validation-based selection of shrinkage parameter in the bayesmixsurv model.*

Description

bayesmixsurv.crossval calculates cross-validation-based, out-of-sample log-likelihood of a bsgw model for a data set, given the supplied folds. bayesmixsurv.crossval.wrapper applies bayesmixsurv.crossval to a set of combinations of shrinkage parameters (lambda1,lambda2) and produces the resulting vector of log-likelihood values as well as the specific combination of shrinkage parameters associated with the maximum log-likelihood. bayesmixsurv.generate.folds generates random partitions, while bayesmixsurv.generate.folds.eventbalanced generates random partitions with events evenly distributed across partitions. The latter feature is useful for cross-valiation of small data sets with low event rates, since it prevents over-accumulation of events in one or two partitions, and lack of events altogether in other partitions.

Usage

```
bayesmixsurv.generate.folds(ntot, nfold=5)
bayesmixsurv.generate.folds.eventbalanced(formula, data, nfold=5)
bayesmixsurv.crossval(data, folds, all=FALSE, print.level=1
  , control=bayesmixsurv.control(), ...)
bayesmixsurv.crossval.wrapper(data, folds, all=FALSE, print.level=1
  , control=bayesmixsurv.control(), lambda.min=0.01, lambda.max=100, nlambda=10
 , lambda1.vec=exp(seq(from=log(lambda.min), to=log(lambda.max), length.out = nlambda))
  , lambda2.vec=NULL
 , lambda12=if (is.null(lambda2.vec)) cbind(lambda1=lambda1.vec, lambda2=lambda1.vec)
  else as.matrix(expand.grid(lambda1=lambda1.vec, lambda2=lambda2.vec)), plot=TRUE, ...)
```

Arguments

ntot	Number of observations to create partitions for. It must typically be set to nrow(data).
nfold	Number of folds or partitions to generate.
formula	Formula specifying the covariates to be used in component 1, and the time/status response variable in the survival model.
data	Data frame containing the covariates and response, used in training and prediction.
folds	An integer vector of length nrow(data), defining fold/partition membership of each observation. For example, in 5-fold cross-validation for a data set of 200 observations, folds must be a 200-long vector with elements from the set {1,2,3,4,5}. Convenience functions bayesmixsurv.generate.folds and bayesmixsurv.generate.folds.eventbalanced can be used to generate the folds vector for a given survival data frame.
all	If TRUE, estimation objects from each cross-validation task is collected and returned for diagnostics purposes.
print.level	Verbosity of progress report.
control	List of control parameters, usually the output of bayesmixsurv.control.
lambda.min	Minimum value used to generate lambda.vec.
lambda.max	Maximum value used to generate lambda.vec.
nlambda	Length of lambda.vec vector.
lambda1.vec	Vector of shrinkage parameters to be tested for component-1 coefficients.
lambda2.vec	Vector of shrinkage parameters to be tested for component-2 coefficients.
lambda12	A data frame that enumerates all combinations of lambda1 and lambda2 to be tested. By default, it is constructed from forming all permutations of lambda1.vec and lambda2.vec. If lambda2.vec=NULL, it will only try equal values of the two parameters in each combination.
plot	If TRUE, and if the lambda1 and lambda2 entries in lambda12 are identical, a plot of loglike as a function of either vector is produced.
...	Further arguments passed to bayesmixsurv.

Value

Functions bayesmixsurv.generate.folds and bayesmixsurv.generate.folds.eventbalanced produce integer vectors of length ntot or nrow(data) respectively. The output of these functions can be directly passed to bayesmixsurv.crossval or bayesmixsurv.crossval.wrapper. Function bayesmixsurv.crossval returns the log-likelihood of data under the assumed bsgw model, calculated using a cross-validation scheme with the supplied fold parameter. If all=TRUE, the estimation objects for each of the nfold estimation jobs will be returned as the "estobjs" attribute of the returned value. Function bayesmixsurv.crossval.wrapper returns a list with elements lambda1 and lambda2, the optimal shrinkage parameters for components 1 and 2, respectively. Additionally, the following attributes are attached:

loglike.vec	Vector of log-likelihood values, one for each tested combination of lambda1 and lambda2.
loglike.opt	The maximum log-likelihood value from the loglike.vec.
lambda12	Data frame with columns lambda1 and lambda2. Each row of this data frame contains one combination of shrinkage parameters that are tested in the wrapper function.
estobjs	If all=TRUE, a list of length nrow(lambda12) is returned, with each element being itself a list of nfold estimation objects associated with each call to the bayesmixsurv function. This object can be examined by the user for diagnostic purposes, e.g. by applying plot against each object.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

Examples

```
# NOTE: to ensure convergence, typically more than 30 samples are needed
folds <- bayesmixsurv.generate.folds.eventbalanced(Surv(futime, fustat) ~ 1, ovarian, 5)
cv <- bayesmixsurv.crossval(ovarian, folds, formula1=Surv(futime, fustat) ~ ecog.ps + rx
  , control=bayesmixsurv.control(iter=30, nskip=10), print.level = 3)
cv2 <- bayesmixsurv.crossval.wrapper(ovarian, folds, formula1=Surv(futime, fustat) ~ ecog.ps + rx
  , control=bayesmixsurv.control(iter=30, nskip=10)
  , lambda1.vec=exp(seq(from=log(0.1), to=log(1), length.out = 3)))
```

---

plot.bayesmixsurv *Plot diagnostics for a bayesmixsurv object*

Description

Four sets of MCMC diagnostic plots are currently generated: 1) log-likelihood trace plots, 2) coefficient trace plots, 3) coefficient autocorrelation plots, 4) coefficient histograms.

Usage

```
## S3 method for class 'bayesmixsurv'
plot(x, pval=0.05, burnin=round(x$control$iter/2), nrow=2, ncol=3, ...)
```

Arguments

x	A bayesmixsurv object, typically the output of bayesmixsurv function.
pval	The P-value at which lower/upper bounds on coefficients are calculated and overlaid on trace plots and historgrams.
burnin	Number of samples discarded from the beginning of an MCMC chain, after which parameter quantiles are calculated.
nrow	Number of rows of subplots within each figure, applied to plot sets 2-4.
ncol	Number of columns of subplots within each figure, applied to plot sets 2-4.
...	Further arguments to be passed to/from other methods.

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

Examples

```
est <- bayesmixsurv(Surv(futime, fustat) ~ ecog.ps + rx, ovarian
            , control=bayesmixsurv.control(iter=800, nskip=100))
plot(est)
```

---

predict.bayesmixsurv *Predict method for bayesmixsurv model fits*

Description

Calculates log-likelihood and hazard/cumulative hazard/survival functions over a user-supplied vector time values, based on bayesmixsurv model object.

Usage

```
## S3 method for class 'bayesmixsurv'
predict(object, newdata=NULL, tvec=NULL, burnin=object$control$burnin, ...)
## S3 method for class 'predict.bayesmixsurv'
summary(object, idx=1:dim(object$smp$h)[3], burnin=object$burnin, pval=0.05
  , popmean=identical(idx,1:dim(object$smp$h)[3]), make.plot=TRUE, ...)
```

Arguments

object	For predict.bayesmixsurv, an object of class "bayesmixsurv", usually the result of a call to bayesmixsurv; for summary.predict.bayesmixsurv, an object of class "predict.bayesmixsurv", usually the result of a call to predict.bayesmixsurv.
newdata	An optional data frame in which to look for variables with which to predict. If omiited, the fitted values (training set) are used.

tvec	An optional vector of time values, along which time-dependent entities (hazard, cumulative hazard, survival) will be predicted. If omitted, only the time-independent entities (currently only log-likelihood) will be calculated. If a single integer is provided for tvec, it is interpreted as number of time points, equally spaced from 0 to object$tmax: tvec <- seq(from=0.0, to=object$tmax, length.out=tvec).
burnin	Number of samples to discard from the beginning of each MCMC chain before calculating median value(s) for time-independent entities.
idx	Index of observations (rows of newdata or training data) for which to generate summary statistics. Default is the entire data.
pval	Desired p-value, based on which lower/upper bounds will be calculated. Default is 0.05.
popmean	Whether population averages must be calculated or not. By default, population averages are only calculated when the entire data is included in prediction.
make.plot	Whether population mean and other plots must be created or not.
...	Further arguments to be passed to/from other methods.

Details

The time-dependent predicted objects (except loglike) are three-dimensional arrays of size (nsmp x nt x nobs), where nsmp = number of MCMC samples, nt = number of time values in tvec, and nobs = number of rows in newdata. Therefore, even for modest data sizes, these objects can occupy large chunks of memory. For example, for nsmp=1000, nt=100, nobs=1000, the three objects h, H, S have a total size of 2.2GB. Since applying quantile to these arrays is time-consuming (as needed for calculation of median and lower/upper bounds), we have left such summaries out of the scope of predict function. Users can instead apply summary to the prediction object to obtain summary statistics. During cross-validation-based selection of shrinkage parameter lambda, there is no need to supply tvec since we only need the log-likelihood value. This significantly speeds up the parameter-tuning process. The function summary.predict.bayesmixsurv allows the user to calculates summary statistics for a subset (or all of) data, if desired. This approach is in line with the overall philosophy of delaying the data summarization until necessary, to avoid unnecessary loss in accuracy due to premature blending of information contained in individual samples.

Value

The function predict.bayesmixsurv returns as object of class "predict.bayesmixsurv" with the following fields:

tvec	Actual vector of time values (if any) used for prediction.
burnin	Same as input.
median	List of median values for predicted entities. Currently, only loglike is produced. See 'Details' for explanation.
smp	List of MCMC samples for predicted entities. Elements include h1,h2,h (hazard functions for components 1,2 and their sum), H1,H2,H (cumulative hazard functions for components 1,2 and their sum), S (survival function), and loglike (model log-likelihood). All functions are evaluated over time values specified in tvec.

km.fit	Kaplan-Meyer fit of the data used for prediction (if data contains response fields).

The function summary.predict.bayesmixsurv returns a list with the following fields:

lower	A list of lower-bound values for h, H, S, hr (hazard ratio of idx[2] to idx[1] observation), and S.diff (survival probability of idx[2] minus idx[1]). The last two are only included if length(idx)==2.
median	List of median values for same entities described in lower.
upper	List of upper-bound values for same entities described in lower.
popmean	Lower-bound/median/upper-bound values for population average of survival probability.
km.fit	Kaplan-Meyer fit associated with the prediction object (if available).

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

Examples

```
est <- bayesmixsurv(Surv(futime, fustat) ~ ecog.ps + rx + age, ovarian
            , control=bayesmixsurv.control(iter=400, nskip=100))
pred <- predict(est, tvec=50)
predsumm <- summary(pred, idx=1:10)
```

---

summary.bayesmixsurv *Summarizing BayesMixSurv model fits*

Description

summary method for class "bayesmixsurv".

Usage

```
## S3 method for class 'bayesmixsurv'
summary(object, pval = 0.05, burnin = object$control$burnin, ...)
## S3 method for class 'summary.bayesmixsurv'
print(x, ...)
```

Arguments

object	An object of class 'bayesmixsurv', usually the result of a call to bayesmixsurv.
x	An object of class "summary.bayesmixsurv", usually the result of a call to summary.bayesmixsurv.
pval	Desired p-value, based on which lower/upper bounds will be calculated. Default is 0.05.
burnin	Number of samples to discard from the beginning of each MCMC chain before calculating median and lower/upper bounds.
...	Further arguments to be passed to/from other methods.

Value

An object of class `summary.bayesmixsurv`, with the following elements:

`call`	The matched call.
`pval`	Same as input.
`burnin`	Same as input.
`single`	Copied from `object$control$single`. See bayesmixsurv.control for explanation.
`coefficients`	A list including matrices `alpha`, `beta1`, `beta2`, and `gamma` (if stratification is used). Each matrix has columns named 'Estimate', 'Lower Bound', 'Upper Bound', and 'P-val'. `alpha` has two rows, one for each components, while each of `beta1` and `beta2` has one row per covariate. `gamma` has one row per stratum (except for the reference stratum).

Author(s)

Alireza S. Mahani, Mansour T.A. Sharabiani

See Also

See summary for a description of the generic method.

The model fitting function is bayesmixsurv.

Examples

```
est <- bayesmixsurv(Surv(futime, fustat) ~ ecog.ps + rx, ovarian
           , control=bayesmixsurv.control(iter=800, nskip=100))
summary(est, pval=0.1)
```

Index