Package 'projpred'

April 16, 2018

Title Projection Predictive Feature Selection

Version 0.8.0

Author Markus Paasiniemi [cre,aut],
Juho Piironen [aut],
Aki Vehtari [aut],
Jonah Gabry [ctb]

Maintainer Juho Piironen <juho.piironen@aalto.fi>

Description Performs projection predictive feature selection for generalized linear models (see, e.g., Piironen and Vehtari, 2017, <doi:10.1007/s11222-016-9649-y>). The package is compatible with 'rstanarm' package, but other reference models can also be used. See the package vignette for more information and examples.

Depends R (>= 3.1.2)

Imports rstan, rstanarm (>= 2.17.4), loo (>= 2.0.0), ggplot2, Rcpp, utils, rngtools (>= 1.2.4)

LinkingTo Rcpp, RcppArmadillo

License GPL-3

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown, glmnet, bayesplot (>= 1.5.0)

VignetteBuilder knitr

URL https://github.com/stan-dev/projpred, http://discourse.mc-stan.org/

BugReports https://github.com/stan-dev/projpred/issues

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-04-16 16:11:31 UTC

R topics documented:

`cvind` *Create cross-validation indices*

Description

Divide indices from 1 to `n` into subsets for `k`-fold cross validation. This function is potentially useful for creating the cross-validation objects for init_refmodel.

Usage

```
cvind(n, k, out = "foldwise", seed = NULL)
```

Arguments

`n`	Number of data points.
`k`	Number of folds.
`out`	Format of the output, either 'foldwise' (default) or 'indices'. See below for details.
`seed`	Random seed so that the same division could be obtained again if needed.

Value

If `out` is 'foldwise', the returned value is a list with `k` elements, each having fields `tr` and `ts` which give the training and test indices, respectively, for each fold. If `out` is 'indices', the returned value is a list with fields `tr` and `ts` each of which is a list with `k` elements giving the training and test indices for each fold.

Examples

```
### compute sample means within each fold
n <- 100
y <- rnorm(n)
cv <- cvind(n, k=5)
cvmeans <- lapply(cv, function(fold) mean(y[fold$itr]))
```

`cv_varsel` *Cross-validate the variable selection (varsel)*

Description

Perform cross-validation for the projective variable selection for a generalized linear model.

Usage

```
cv_varsel(fit, method = NULL, cv_method = NULL, ns = NULL, nc = NULL,
  nspred = NULL, ncpred = NULL, nv_max = NULL, intercept = NULL,
  penalty = NULL, verbose = T, nloo = 100, K = NULL, k_fold = NULL,
  lambda_min_ratio = 1e-05, nlambda = 500, regul = 1e-06,
  validate_search = T, seed = NULL, ...)
```

Arguments

`fit`	Same as in varsel.
`method`	Same as in varsel.
`cv_method`	The cross-validation method, either 'LOO' or 'kfold'. Default is 'LOO'.
`ns`	Number of samples used for selection. Ignored if nc is provided or if method='L1'.
`nc`	Number of clusters used for selection. Default is 1 and ignored if method='L1' (L1-search uses always one cluster).
`nspred`	Number of samples used for prediction (after selection). Ignored if ncpred is given.
`ncpred`	Number of clusters used for prediction (after selection). Default is 5.
`nv_max`	Same as in varsel.
`intercept`	Same as in varsel.
`penalty`	Same as in varsel.
`verbose`	Whether to print out some information during the validation, Default is TRUE.
`nloo`	Number of observations used to compute the LOO validation (anything between 1 and the total number of observations). Smaller values lead to faster computation but higher uncertainty (larger errorbars) in the accuracy estimation. Default value is 100. Only applicable if `cv_method = LOO`.

K	Number of folds in the k-fold cross validation. Only applicable if cv_method = TRUE and k_fold = NULL.
k_fold	An array with cross-validated stanfits and the respective test datasets returned by kfold with save_fits=TRUE. If not provided, kfold is called inside the function.
lambda_min_ratio	Same as in varsel.
nlambda	Same as in varsel.
regul	Amount of regularization in the projection. Usually there is no need for regularization, but sometimes for some models the projection can be ill-behaved and we need to add some regularization to avoid numerical problems. Default is 1e-9.
validate_search	Whether to cross-validate also the selection process, that is, whether to perform selection separately for each fold. Default is TRUE and we strongly recommend not setting this to FALSE, because this is known to bias the accuracy estimates for the selected submodels. However, setting this to FALSE can sometimes be useful because comparing the results to the case where this parameter is TRUE gives idea how strongly the feature selection is (over)fitted to the data (the difference corresponds to the search degrees of freedom or the effective number of parameters introduced by the selectin process).
seed	Random seed used in the subsampling LOO. By default uses a fixed seed.
...	Currently ignored.

Value

The original stanreg object augmented with an element 'varsel', which is a list containing the following elements:

vind The order in which the variables were added to the submodel.

pctch Percentage of cross-validation runs that included the given variable to a model of given size.

kl KL-divergence for each submodel size.

summaries CV-summary statistics computed during the selection.

d_test The data used to evaluate the summaries.

family_kl A modified family-object.

Examples

```
### Usage with stanreg objects
fit <- stan_glm(y~x, binomial())
fit_cv <- cv_varsel(fit)
```

df_binom *Binomial toy example.*

Description

Binomial toy example.

Usage

```
df_binom
```

Format

A simulated classification dataset containing 100 observations.

y target, 0 or 1.

x features, 30 in total.

Source

http://web.stanford.edu/~hastie/glmnet/glmnetData/BNExample.RData

df_gaussian *Gaussian toy example.*

Description

Gaussian toy example.

Usage

```
df_gaussian
```

Format

A simulated regression dataset containing 100 observations.

y target, real-valued.

x features, 20 in total. Mean and sd approximately 0 and 1.

Source

http://web.stanford.edu/~hastie/glmnet/glmnetData/QSExample.RData

`init_refmodel` *Generic reference model initialization*

Description

Initializes a structure that can be used as a reference fit for the projective variable selection. This function is provided to allow construction of the reference fit using also other tools than `rstanarm`, because only certain specific information is needed for the actual projection and variable selection.

Usage

```
init_refmodel(x, y, family, predfun = NULL, dis = NULL, offset = NULL,
  wobs = NULL, wsample = NULL, intercept = TRUE, cvfits = NULL)
```

Arguments

`x`	Predictor matrix of dimension `n`-by-`D` containing the candidate variables for selection (i.e. variables from which to select the submodel). Rows denote the observations and columns the different variables.
`y`	Vector of length `n` giving the target variable values.
`family`	family object giving the model family
`predfun`	Function that takes a `nt`-by-`D` test predictor matrix as an input (`nt` = # test points, `D` = # predictors) and outputs a `nt`-by-`S` matrix of expected values for the target variable y, each column corresponding to one posterior draw for the parameters in the reference model (the number of draws `S` can also be 1). The output should be computed without any offsets, these are automatically taken into account internally, e.g. in cross-validation.
`dis`	Vector of length `S` giving the posterior draws for the dispersion parameter in the reference model if there is such a parameter in the model family. For Gaussian observation model this is the noise std `sigma`.
`offset`	Offset to be added to the linear predictor in the projection. (Same as in function `glm`.)
`wobs`	Observation weights. If omitted, equal weights are assumed.
`wsample`	vector of length `S` giving the weights for the posterior draws. If omitted, equal weights are assumed.
`intercept`	Whether to use intercept. Default is `TRUE`.
`cvfits`	A list with K elements, each of which is a list with fields including at least variables: tr, ts and predfun giving the training and test indices and prediction function for each fold. Additionally each element can have field dis (dispersion samples for each fold) if the model has a dispersion parameter. Can be omitted but needed for K-fold cross validation for genuine reference models.

Value

An object that can be passed to all the functions that take the reference fit as the first argument, such as varsel, cv_varsel, proj_predict and proj_linpred.

proj-pred *Extract draws of the linear predictor and draw from the predictive distribution of the projected submodel*

Description

proj_linpred extracts draws of the linear predictor and proj_predict draws from the predictive distribution of the projected submodel or submodels. If the projection has not been performed, the functions also perform the projection.

Usage

```
proj_linpred(object, xnew, ynew = NULL, offsetnew = NULL,
  weightsnew = NULL, nv = NULL, transform = FALSE, integrated = FALSE,
  ...)

proj_predict(object, xnew, offsetnew = NULL, weightsnew = NULL, nv = NULL,
  draws = NULL, seed_samp = NULL, ...)
```

Arguments

object	The object returned by varsel, cv_varsel or project.
xnew	The predictor values used in the prediction. If vind is specified, then xnew should either be a dataframe containing column names that correspond to vind or a matrix with the number and order of columns corresponding to vind. If vind is unspecified, then xnew must either be a dataframe containing all the column names as in the original data or a matrix with the same columns at the same positions as in the original data.
ynew	New (test) target variables. If given, then the log predictive density for the new observations is computed.
offsetnew	Offsets for the new observations. By default a vector of zeros.
weightsnew	Weights for the new observations. For binomial model, corresponds to the number trials per observation. For proj_linpred, this argument matters only if ynew is specified. By default a vector of ones.
nv	Number of variables in the submodel (the variable combination is taken from the varsel information). If a list, then results for all specified model sizes are returned. Ignored if vind is specified.
transform	Should the linear predictor be transformed using the inverse-link function? Default is FALSE. For proj_linpred only.
integrated	If TRUE, the output is averaged over the parameters. Default is FALSE. For proj_linpred only.
...	Additional argument passed to project if object is an object returned by varsel or cv_varsel.
draws	Number of draws to return from the predictive distribution of the projection. The default is 1000. For proj_predict only.

seed_samp — An optional seed to use for drawing from the projection. For proj_predict only.

Value

If the prediction is done for one submodel only (nv has length one or vind is specified) and ynew is unspecified, a matrix or vector of predictions (depending on the value of integrated). If ynew is specified, returns a list with elements pred (predictions) and lpd (log predictive densities). If the predictions are done for several submodel sizes, returns a list with one element for each submodel.

---

project — *Projection to submodels of selected sizes*

Description

Perform the projection predictive variable selection for a generalized linear model fitted with rstanarm.

Usage

```
project(object, nv = NULL, vind = NULL, ns = NULL, nc = NULL,
  intercept = NULL, seed = NULL, regul = 1e-06, ...)
```

Arguments

object	The object returned by varsel or cv_varsel.
nv	Number of variables in the submodel (the variable combination is taken from the varsel information). If a list, then the projection is performed for each model size. Default is the model size suggested by the variable selection (see function suggest_size). Ignored if vind is specified.
vind	Variable indices onto which the projection is done. If specified, nv is ignored.
ns	Number of samples to be projected. Ignored if nc is specified. Default is 400.
nc	Number of clusters in the clustered projection.
intercept	Whether to use intercept. Default is TRUE.
seed	A seed used in the clustering (if nc!=ns). Can be used to ensure same results every time.
regul	Amount of regularization in the projection. Usually there is no need for regularization, but sometimes for some models the projection can be ill-behaved and we need to add some regularization to avoid numerical problems. Default is 1e-9.
...	Currently ignored.

Value

A list of submodels (or a single submodel if projection was performed onto a single variable combination), each of which contain the following elements:

`kl` The kl divergence from the full model to the submodel.

`weights` Weights for each draw of the projected model.

`dis` Draws from the projected dispersion parameter.

`alpha` Draws from the projected intercept.

`beta` Draws from the projected weight vector.

`vind` The order in which the variables were added to the submodel.

`intercept` Whether or not the model contains an intercept.

`family_kl` A modified `family`-object.

Examples

```
### Usage with stanreg objects
fit <- stan_glm(y~x, binomial())
fit_v <- varsel(fit)
proj4 <- project(fit_v, nv = 4)
```

`projpred` *Projection predictive feature selection*

Description

Description

projpred is an R package to perform projection predictive variable selection for generalized linear models. The package is aimed to be compatible with **rstanarm** but also other reference models can be used (see function `init_refmodel`).

Currently, the supported models (family objects in R) include Gaussian, Binomial and Poisson families, but more will be implemented later. See the quickstart-vignette for examples.

Functions

varsel, cv_varsel, init_refmodel Perform and cross-validate the variable selection. init_refmodel can be used to initialize a reference model other than **rstanarm**-fit.

project Get the projected posteriors of the reduced models.

proj_predict, proj_linpred Make predictions with reduced number of features.

varsel_plot, varsel_stats Visualize and get some key statistics about the variable selection.

References

Dupuis, J. A. and Robert, C. P. (2003). Variable selection in qualitative models via an entropic explanatory power. *Journal of Statistical Planning and Inference*, 111(1-2):77–94.

Goutis, C. and Robert, C. P. (1998). Model choice in generalised linear models: a Bayesian approach via Kullback–Leibler projections. *Biometrika*, 85(1):29–37.

Juho Piironen and Aki Vehtari (2017). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711-735. doi:10.1007/s11222-016-9649-y. (Online).

`suggest_size` *Suggest model size*

Description

This function can be used for suggesting an appropriate model size based on certain rule. Notice that the decision rules are heuristic and should be interpreted as guidelines. It is recommended that the user studies the results via `varsel_plot` and or `varsel_stats` and makes the final decision based on what is most appropriate for the given problem.

Usage

```
suggest_size(object, stat = "elpd", alpha = 0.32, pct = 0,
  type = "upper", warnings = TRUE, ...)
```

Arguments

`object`	The object returned by varsel or cv_varsel.
`stat`	Statistic used for the decision. Default is elpd. See `varsel_stats` for other possible choices.
`alpha`	A number indicating the desired coverage of the credible intervals based on which the decision is made. E.g. `alpha=0.1` corresponds to 90% probability mass within the intervals. See details for more information.
`pct`	Number indicating the relative proportion between full model and null model utilities one is willing to sacrifice. See details for more information.
`type`	Either 'upper' (default) or 'lower' determining whether the decisions are based on the upper or lower credible bounds. See details for more information.
`warnings`	Whether to give warnings if automatic suggestion fails, mainly for internal use. Default is TRUE, and usually no reason to set to FALSE.
`...`	Currently ignored.

Details

The suggested model size is the smallest model for which either the lower or upper (depending on argument `type`) credible bound of the submodel utility u_k with significance level `alpha` falls above

$$u_ref - pct * (u_ref - u_0)$$

Here u_ref denotes the reference model utility and u_0 the null model utility (currently the utility is taken to be the mean log predictive density, MLPD). The lower and upper bounds are defined to contain the submodel utility with probability 1-alpha (each tail has mass alpha/2).

By default `ratio=0`, `alpha=0.32` and `type='upper'` which means that we select the smallest model for which the upper tail exceeds the reference model level, that is, which is better than the reference model with probability 0.16 (and consequently, worse with probability 0.84). In other words, the estimated difference between the reference model and submodel utitlities is at most one standard error away from zero, so the two utilities are considered to be close.

`varsel` *Variable selection for generalized linear models*

Description

Perform the projection predictive variable selection for a generalized linear model fitted with rstanarm.

Usage

```
varsel(fit, d_test = NULL, method = NULL, ns = NULL, nc = NULL,
  nspred = NULL, ncpred = NULL, nv_max = NULL, intercept = NULL,
  penalty = NULL, verbose = F, lambda_min_ratio = 1e-05, nlambda = 500,
  regul = 1e-06, ...)
```

Arguments

`fit`	Either a stanreg-object or an object returned by init_refmodel.
`d_test`	A test dataset, which is used to evaluate model performance. If not provided, training data is used. Currently this argument is for internal use only.
`method`	The method used in the variable selection. Possible options are `'L1'` for L1-search and `'forward'` for forward selection. Default is 'forward' if the number of variables in the full data is at most 20, and `'L1'` otherwise.
`ns`	Number of posterior draws used in the variable selection. Cannot be larger than the number of draws in the full model. Ignored if nc is set.
`nc`	Number of clusters to use in the clustered projection. Overrides the `ns` argument. Defaults to 1.
`nspred`	Number of samples used for prediction (after selection). Ignored if ncpred is given.
`ncpred`	Number of clusters used for prediction (after selection). Default is 5.

`nv_max` Maximum number of varibles until which the selection is continued. Defaults to min(20, D, floor(0.4*n)) where n is the number of observations and D the number of variables.

`intercept` Whether to use intercept in the submodels. Defaults to TRUE.

`penalty` Vector determining the relative penalties or costs for the variables. Zero means that those variables have no cost and will therefore be selected first, whereas Inf means that those variables will never be selected. Currently works only if method == 'L1'. By default 1 for each variable.

`verbose` If TRUE, may print out some information during the selection. Defaults to FALSE.

`lambda_min_ratio` Ratio between the smallest and largest lambda in the L1-penalized search. This parameter essentially determines how long the search is carried out, i.e., how large submodels are explored. No need to change the default value unless the program gives a warning about this.

`nlambda` Number of values in the lambda grid for L1-penalized search. No need to change unless the program gives a warning about this.

`regul` Amount of regularization in the projection. Usually there is no need for regularization, but sometimes for some models the projection can be ill-behaved and we need to add some regularization to avoid numerical problems. Default is 1e-9.

`...` Currently ignored.

Value

The original fit-object object augmented with a field 'varsel', which is a list containing the following elements:

`vind` The order in which the variables were added to the submodel.

`kl` KL-divergence for each submodel size.

`summaries` Summary statistics computed during the selection.

`d_test` The data used to evaluate the summaries.

`family_kl` A modified family-object.

Examples

```
### Usage with stanreg objects
fit <- stan_glm(y~x, binomial())
fit_v <- varsel(fit)
plot_varsel(fit_v)
```

varsel-statistics *Plot or fetch summary statistics related to variable selection*

Description

varsel_stats can be used to obtain summary statistics related to variable selection. The same statistics can be plotted with varsel_plot.

Usage

```
varsel_plot(object, nv_max = NULL, stats = "elpd", deltas = F,
  alpha = 0.1, ...)

varsel_stats(object, nv_max = NULL, stats = "elpd", type = c("mean",
  "se"), deltas = F, alpha = 0.1, ...)
```

Arguments

object	The object returned by varsel or cv_varsel.
nv_max	Maximum submodel size for which the statistics are calculated.
stats	One or several strings determining which statistics to calculate. Available statistics are: • elpd: (Expected) sum of log predictive densities • mlpd: Mean log predictive density, that is, elpd divided by the number of datapoints. • mse: Mean squared error (gaussian family only) • rmse: Root mean squared error (gaussian family only) • acc/pctcorr: Classification accuracy (binomial family only) Default is elpd.
deltas	If TRUE, the difference between the full model and the submodel is returned instead of the actual value of the statistic. Defaults to FALSE.
alpha	A number indicating the desired coverage of the credible intervals. E.g. alpha=0.1 corresponds to 90% probability mass within the intervals.
...	Currently ignored.
type	One or more items from 'mean', 'se', 'lower' and 'upper' indicating which of these to compute (mean, standard error, and lower and upper credible bounds). The credible bounds are determined so that 1-alpha percent of the mass falls between them.

Index