k-fold cross-validation

May 25, 2005

R topics documented:

---

kxvglm

Description

Performs k-fold cross-validation of a fixed-effects *present* / *available* probability model.

Usage

```
kxvglm(formula, data, k=5, nbin=10, presval=1, partition=NULL, family =
binomial(link="logit"), ... )
```

Arguments

`formula`	a model with a binary response variable (denoting *present* or *available*). The model must be in a form acceptable to the `glm` function.
`data`	the dataframe on which to fit the model
`k`	the number of equal-sized subsets into which the *present* observations should be randomly partitioned. If k does not evenly divide into the number of *present* rows in `data`, then `kxvglm` does the best it can: some of the `k` subsets will have exactly one more observation than the others do.
`presval`	the value of the response variable corresponding to *present* (rather than *available*) observations
`nbin`	the number of bins into which to divide the modelled values
`partition`	a character string naming the column in `data` by which the *present* data should be partitioned. The column is treated as a factor, with each level providing a different subset of the *present* data. If `partition` is specified, then `k` is ignored.
`family`	the link/error function family for `glm`
`...`	additional parameters passed to `glm`

Details

The dataframe consists of *present* and *available* observations, determined by the value of the response variable (i.e. left-hand side) in `formula`. Observations where the response variable has the value `presval` are *present*, all others are *available*.

`kxvglm` partitions the *present* observations of `data`, either randomly into `k` equal-sized (if possible) subsets or by levels of the column `partition`, if specified. For each subset `S` in the partition, a model is fitted to the reduced dataset `data - S`, and this is used to predict relative probability of *presence* for observations in `S`. The function `binrank` is called to test (and optionally plot) the association between frequency of *present* observations and model predictions.

For datasets with clumps (i.e. large numbers of observations having the same values for all the variables in the model), the model output will have a spikey distribution which may prevent its division into well-defined bins. In this case, you can either reduce `nbin`, or have `kxvglm` add a zero-mean normally-distributed *fuzz* to each modelled value by setting the global variable `kxvFuzzFactor` to a small positive number (e.g. `kxvFuzzFactor <- 0.001`). See `binrank` for details.

Value

A table with one row for each subset in the partition (e.g. `1..k`), and a summary *meanfreq* row. The table rows give the subset name, the Spearman correlation coefficient between *presence* frequency and bin rank, and its p-value. A *meanfreq* entry is appended, which gives the results of testing the mean bin *presence* frequency against bin rank (i.e. it is not the mean of the other entries in the table, but the result of testing the association shown in the second plot, described below).

Side-effects

If the global variable `kxvPrintFlag` is `TRUE`, then a summary of each fitted model is printed.

If the global variable `kxvPlotFlag` is `TRUE`, then a page with two plots is produced. The first shows the area-adjusted proportion of *present* observations within bins, and has a curve for each subset in the partition. The second shows the mean (+/- SD) of the values in each bin of the first plot.

Warning

The fuzzy binning obtained by setting `kxvFuzzFactor` to a positive value is an unholy non-peer-reviewed fix. Do not use it without consulting a statistician!

Author(s)

John Brzustowski ⟨jbrzusto@fastmail.fm⟩

References

Mark S. Boyce et al. (2002) Evaluating resource selection functions. *Ecological Modelling* 157 Pages 281–300.

See Also

This function calls `kxv` and `binrank`. To find clumps in your data, which might generate binning errors, see `clumps`.

Examples

```
elk <- read.csv(file="HR_Summer.csv", header=T)
kxvglm( PTTYPE~DEM30M+DEM30M^2+VARCES1+FORGSUM+FORGSUM^2, elk, k=5, nbin=10)
```

---

kxv

Description

Performs k-fold cross-prediction of a *present* / *available* probability model. (This is a template function that does **not** perform cross-validation; see `kxvglm` for that.)

Usage

```
kxv(present, k=5, partition = NULL, data, modfun, predfun)
```

Arguments

`present` a logical vector with one element per observation in `data`. TRUE represents *present* and FALSE represents *available*.

`k` the number of equal-sized subsets into which the *present* data should be randomly partitioned. If k does not evenly divide into the number of *present* observations in `data`, then `kxv` does the best it can: some of the `k` subsets will have exactly one more observation than the others do.

`partition` a character string naming the column in `data` by which the *present* data should be partitioned. The column is treated as a factor, with each level providing a different subset of the *present* data. If `partition` is specified, then `k` is ignored.

`data` a data frame to be partitioned and modelled.

`modfun` a function which fits a model to a dataframe and returns the model object as its result

`predfun` a function which predicts model values for a dataframe. It must accept two unnamed positional parameters: the model returned by `modfun`, and the (sub) dataframe for which to predict *present* by using the model. The return value is usually a numeric vector with the same length as the number of observations in the (sub) dataframe. It will typically be the fitted relative probability of *present* for each observation.

Details

`kxv` partitions the *present* observations of `data`, either randomly into `k` equal-sized (if possible) subsets or by levels of the specified factor column. For each subset `S` in the partition, a model is fitted to the reduced dataset `data - S`, and then used to predict relative probability of *presence* for observations in `S`. This function will usually be called indirectly by use of a convenience function such as `kxvglm`. If the global variable `kxvPrintFlag` is `TRUE`, a summary of each fitted model is printed.

Value

A list with the same length as the number of subsets in the partition (e.g. `k`). Each element in the list corresponds to a subset, `S`, of the partition, and is itself a list with these two components:

`model`	the model fitted to `data - S`
`pred`	the values predicted by the model for `S`

The elements of the list are indexed by `1..k` or by the levels of the factor `partition`. Output from this function is typically fed into a comparison function such as `binrank`

Author(s)

John Brzustowski ⟨jbrzusto@fastmail.fm⟩

References

Mark S. Boyce et al. (2002) Evaluating resource selection functions. *Ecological Modelling* 157 Pages 281–300.

See Also

The convenience function `kxvglm`. The output from `kxv` is more useful when fed into a function like like `binrank`.

binrank

Description

Process the output of `kxv` by binning modelled values for *available* observations, and then testing the association between frequency of *present* observations and bin ranks by Spearman's rank correlation coefficient test. Optionally plot area-adjusted frequency of *present* vs. bin rank.

Usage

```
binrank(out, nbin=10, fitted, subnames=NULL, fuzzfactor=0)
```

Arguments

`out`	the output from the `kxv` function
`nbin`	the number of bins into which to divide the modelled values
`fitted`	a function which extracts the fitted model values for *available* observations from the model objects in `out`; i.e. `fitted(out[[1]][["model"]])` should be the fitted values for *available* observations using the model which was fitted to the data excluding the first subset.
`subnames`	a character vector of names of the subsets which are excluded, one at a time, in fitting the models. If `out` was generated by calling `kxv` with a `partition` parameter, then `subnames` should be the names of its levels. If random k-fold partitioning was used, then `subnames` should be left `NULL`, which will prevent a (useless) legend from being shown on the first plot.

`fuzzfactor` a non-negative number used for fuzzy binning. If this is positive, it is used to add a normally-distributed zero-mean random error term to each modelled value before binning. For each fitted model, the *fuzz* is computed as `fuzzfactor` times the mean magnitude of all the fitted and predicted model values. A random normal deviate `~N(0, fuzz)` is added to each fitted or predicted model value before calculating bin boundaries and bin frequencies.

This may be useful when a model has clumps (many duplicated values), but is a use-at-your-own-risk feature.

Details

`binrank` accepts output from `kxv` and for each excluded subset, S, assesses how well the model fitted to D-S predicts S. The method is:

- take the fitted values of *available* observations under the model built using D-S, and divide them into `nbin` equal-sized (or as near as possible) bins.
- tally the proportion of observations in S which are in each bin
- scale the proportions so that 1 represents what would be expected if *present* observations were randomly assigned to bins. In keeping with *Boyce et al. (2002)*, this is called the *area-adjusted frequency*.
- perform a Spearman rank correlation test between scaled proportions and bin ranks

Value

A table with an entry for each partition in the `kxv` output, giving the correlation coefficient and p-value for the correlation test between scaled proportions and bin ranks. Also, a *mean* entry is appended, which gives the results of testing the mean scaled bin proportions against bin rank (i.e. it is not the mean of the other entries in the table). High values of the correlation coefficient indicate the model passed to `kxv` is generating higher values for *present* observations than for *available* ones, as one would desire of a valid model.

Side-effects

If the global variable `kxvPlotFlag` is `TRUE`, then a page with two plots is produced. The first shows the area-adjusted proportion of *present* observations within bins, and has a curve for each subset in the partition. The second shows the mean (+/- SD) of the values in each bin of the first plot.

Author(s)

John Brzustowski ⟨jbrzusto@fastmail.fm⟩

References

Mark S. Boyce et al. (2002) Evaluating resource selection functions. *Ecological Modelling* 157 Pages 281–300.

See Also

The convenience function `kxvglm` calls `binrank`. The function `kxv` generates output suitable as input to `binrank`. The function `clumps` may suggest that your model will need fuzzy binning.

clumps

Description

Finds the combinations of predictor and response variable values which occur most frequently in a dataset.

Usage

```
clumps(formula, data, n = 10)
```

Arguments

formula	a formula whose terms are variables in data
data	a dataframe
n	the number of clumps to find

Details

Models with variables having values in a small discrete set will sometimes have many observations with identical values for these variables. This is sometimes the case even when the variable is conceptually continuous, but is estimated with coarse precision. `clumps` finds the `n` most frequent combinations of values in `data` for the variables occurring in `formula`. All variables in the model are used, whether they appear alone in their own term or not. Only values of the variables themselves, and not of any higher-order terms, are considered (but if two observations have identical values for the variables, they will also have identical values for higher-order terms, of course).

Value

A table with `n` rows, listing the values for the variables in each clump, and a column, `(count)`, with the number of observations having that combination of values. The rows are sorted in decreasing order of `(count)`.

Author(s)

John Brzustowski ⟨jbrzusto@fastmail.fm⟩

See Also

The function `binrank` which processes output from `kxvglm`, suggests the user call `clumps` when a model produces too many duplicate values for the desired number of bins.

Index