

Package ‘distrom’

August 29, 2016

Title Distributed Multinomial Regression

Version 0.3-3

Author Matt Taddy <taddy@chicagobooth.edu>

Depends R (>= 2.15), Matrix, gamlr, parallel, methods, stats

Suggests MASS, textir

Description

Estimation for a multinomial logistic regression factorized into independent Poisson log regressions. See the textir package for applications in multinomial inverse regression analysis of text.

Maintainer Matt Taddy <taddy@chicagobooth.edu>

License GPL-3

URL <http://faculty.chicagobooth.edu/matt.taddy/research/index.html>

<http://github.com/TaddyLab/distrom>

NeedsCompilation no

Repository CRAN

Date/Publication 2015-08-19 00:22:27

R topics documented:

collapse	2
dmr	3
dmrcoef-class	5
Index	7

`collapse`*Data checking and binning*

Description

Collapses counts along equal levels of binned covariates.

Usage

```
collapse(v, counts, mu=NULL, bins=NULL)
```

Arguments

<code>v</code>	Either matrix or Matrix of covariates (matches covars in dmr).
<code>counts</code>	Either matrix or Matrix of multinomial counts, or a factor (matches counts in dmr).
<code>mu</code>	Possible pre-specified fixed effects for dmr; otherwise they are calculated here.
<code>bins</code>	The number of quantile bins into which we collapse <code>v</code> . <code>bins=NULL</code> does no collapsing.

Details

For each column of `v`, aggregates the observations into bins defined by their average value. Both `v` and `counts` are then collapsed according to levels of the interaction across implied bin-factors, and the number of observations in each bin is recorded as `n`. Look at the code of the `dmr` function to see `collapse` used in practice.

Value

A list containing collapsed and formatted `v`, `counts`, and `nbin`, along with `mu = log(rowSums(counts))`, the plug-in fixed effect estimates for `dmr`.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

`we8there`

Description

Gamma-lasso path estimation for a multinomial logistic regression factorized into independent Poisson log regressions.

Usage

```
dmr(cl, covars, counts, mu=NULL, bins=NULL, verb=0, cv=FALSE, ...)
## S3 method for class 'dmr'
coef(object, ...)
## S3 method for class 'dmr'
predict(object, newdata,
        type=c("link", "response", "class"), ...)
```

Arguments

<code>cl</code>	A parallel library socket cluster. If <code>is.null(cl)</code> , everything is done in serial. See <code>help(parallel)</code> , <code>help(makeCluster)</code> , and our examples here for details.
<code>covars</code>	A dense matrix or sparse Matrix of covariates. This should not include the intercept.
<code>counts</code>	A dense matrix or sparse Matrix of response counts.
<code>mu</code>	Pre-specified fixed effects for each observation in the Poisson regression linear equation. If <code>mu=NULL</code> , then we use <code>log(rowSums(x))</code> . Note that if <code>bins</code> is non-null then this argument is ignored and <code>mu</code> is recalculated on the collapsed data.
<code>bins</code>	Number of bins into which we will attempt to collapse each column of <code>covars</code> . Since sums of multinomials with equal probabilities are also multinomial, the model is then fit to these collapsed ‘observations’. <code>bins=NULL</code> does no collapsing.
<code>verb</code>	Whether to print some info. <code>max(0, verb-1)</code> is passed on to <code>gamlr</code> and will print if you created an outfile when specifying <code>cl</code> .
<code>cv</code>	A flag for whether to use <code>cv.gamlr</code> instead of <code>gamlr</code> for each Poisson regression.
<code>type</code>	For <code>predict.dmr</code> , this is the scale upon which you want prediction. Under "link", just the linear map <code>newdata times object</code> , under "response" the fitted multinomial probabilities, under "class" the max-probability class label. For sufficient reductions see the <code>srproj</code> function of the <code>textir</code> library.
<code>newdata</code>	A Matrix with the same number of columns as <code>covars</code> .
<code>...</code>	Additional arguments to <code>gamlr</code> , <code>cv.gamlr</code> , and their associated methods.
<code>object</code>	A <code>dmr</code> list of fitted <code>gamlr</code> models for each response category.

Details

dmr fits multinomial logistic regression by assuming that, unconditionally on the ‘size’ (total count across categories) each individual category count has been generated as a Poisson

$$x_{ij} \sim Po(\exp[\mu_i + \alpha_j + \beta v_i]).$$

We [default] plug-in estimate $\hat{\mu}_i = \log(m_i)$, where $m_i = \sum_j x_{ij}$ and p is the dimension of x_i . Then each individual is outsourced to Poisson regression in the `gam1r` package via the `parLapply` function of the `parallel` library. The output from `dmr` is a list of `gam1r` fitted models.

`coef.dmr` builds a matrix of multinomial logistic regression coefficients from the `length(object)` list of `gam1r` fits. Default selection under `cv=FALSE` uses an information criteria via AICc on Poisson deviance for each individual response dimension (see `gam1r`). Combined coefficients across all dimensions are then returned as a `dmrcoef` `s4`-class object.

`predict.dmr` takes either a `dmr` or `dmrcoef` object and returns predicted values for newdata on the scale defined by the `type` argument.

Value

dmr returns the `dmr` `s3` object: an `ncol(counts)`-length list of fitted `gam1r` objects, with the added attributes `nlambda`, `mu`, and `nobs`.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2013) Distributed Multinomial Regression

Taddy (2013) The Gamma Lasso

Taddy (2013) Multinomial Inverse Regression for Text Analysis, with discussion and rejoinder, *Journal of the American Statistical Association*.

See Also

`dmrcoef`-class, `cv.dmr`, `AICc`, and the `gam1r` and `textir` packages.

Examples

```
library(MASS)
data(fgl)

## make your cluster
## FORK is faster but memory heavy, and doesn't work on windows.
cl <- makeCluster(2,type=ifelse(.Platform$OS.type=="unix","FORK","PSOCK"))
print(cl)

## fit in parallel
fits <- dmr(cl, fgl[,1:9], fgl$type, verb=1)
```

```

## its good practice stop the cluster once you're done
stopCluster(cl)

## Individual Poisson model fits and AICc selection
par(mfrow=c(3,2))
for(j in 1:6){
plot(fits[[j]])
mtext(names(fits)[j],font=2,line=2) }

## AICc model selection
B <- coef(fits)

## Fitted probability by true response
par(mfrow=c(1,1))
P <- predict(B, fgl[,1:9], type="response")
boxplot(P[cbind(1:214,fgl$type)]~fgl$type,
ylab="fitted prob of true class")

```

dmrcoef-class	<i>Class "dmrcoef"</i>
---------------	------------------------

Description

The extended dgCMatrix class for output from `coef.dmr`.

Details

This is the class for a covariate matrix from dmr regression; it inherits the dgCMatrix class as defined in the Matrix library. In particular, this is the `ncol(covars)` by `ncol(counts)` matrix of logistic regression coefficients chosen in `coef.dmr` from the regularization paths for each category.

Objects from the Class

Objects can be created only by a call to the `coef.dmr` function.

Slots

i: From dgCMatrix: the row indices.
p: From dgCMatrix: the column pointers.
Dim: From dgCMatrix: the dimensions.
Dimnames: From dgCMatrix: the list of labels.
x: From dgCMatrix: the nonzero entries.
factors: From dgCMatrix.

Extends

Class `dgMatrix`, directly.

Methods

predict signature(object = "dmrcoef"): Prediction for a given dmrcoef matrix. Takes the same arguments as `predict.dmr`, but will be faster (since `coef.dmr` is called inside `predict.dmr`).

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

`dmr`, `coef.dmr`, `predict.dmr`

Examples

```
showClass("dmrcoef")
```

Index

*Topic **classes**

dmrcoef-class, [5](#)

coef.dmr (dmr), [3](#)

collapse, [2](#)

dgCMatrix, [6](#)

distrom (dmr), [3](#)

dmr, [3](#)

dmrcoef-class, [5](#)

predict, dmrcoef-method (dmrcoef-class),
[5](#)

predict.dmr (dmr), [3](#)