Package 'textir'

August 9, 2012

Title Inverse Regression for Text Analysis

Version 1.8-8

Date August 2012

Author Matt Taddy <taddy@chicagobooth.edu>

Depends R (>= 2.10), slam

Suggests MASS

Description A suite of tools for text and sentiment mining. This
includes the 'mnlm' function, for sparse multinomial logistic
regression, 'pls', a concise partial least squares routine, and
the 'topics' function, for efficient estimation and dimension selection in latent topic models.

Maintainer Matt Taddy <taddy@chicagobooth.edu>

License GPL-3

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

References Taddy (2012) ''Multinomial Inverse Regression for Text
Analysis'' (http://arxiv.org/abs/1012.2098) and ''On Estimation
and Selection for Topic Models'' (AISTATS 2012,http://arxiv.org/abs/1109.4518).

Repository CRAN

Date/Publication 2012-08-09 05:35:12

R topics documented:

`textir-package` *Inverse Regression for Text*

Description

Tools for analysis of sentiment in text

Details

Check out Taddy (2011) and the help files.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

http://faculty.chicagobooth.edu/matt.taddy/index.html

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

pls, mnlm

coef.mnlm *mnlm coefficients*

Description

Coefficients for Multinomial Regression

Usage

```
## S3 method for class 'mnlm'
coef( object, origscale=TRUE, ... )
```

Arguments

object	An output object from the pls function.
origscale	Whether to ouput coefficients on the original covariate scale (i.e. before possible normalization). Default is TRUE, and origscale=FALSE just outputs raw loadings for the fitted model
...	Additional unused arguments.

Value

A ncol(object$covars)+1 by ncol(object$counts) (or by 1 for binary response) matrix of coefficients, including the intercept.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

See Also

mnlm

congress109 *Ideology in Political Speeches*

Description

Phrase counts and ideology scores by speaker for members of the 109th US congress.

Details

This data originally appears in Gentzkow and Shapiro (GS; 2010) and considers text of the 2005 Congressional Record, containing all speeches in that year for members of the United States House and Senate. In particular, GS record the number times each of 529 legislators used terms in a list of 1000 phrases (i.e., each document is a year of transcripts for a single speaker). Associated sentiments are repshare – the two-party vote-share from each speaker's constituency (congressional district for representatives; state for senators) obtained by George W. Bush in the 2004 presidential election – and the speaker's first and second common-score values (from http://voteview.com). Full parsing and sentiment details are in Taddy (2011; Section 2.1).

Value

`congress109Counts`
: A `simple_triplet_matrix` of phrase counts indexed by speaker-rows and phrase-columns.

`congress109Ideology`
: A `matrix` containing the associated `repshare` and common scores `[cs1,cs2]`, as well as speaker characteristics: `party` ('R'epublican, 'D'emocrat, or 'I'ndependent), `state`, and `chamber` ('H'ouse or 'S'enate).

Author(s)

Matt Taddy, <taddy@chicagobooth.edu>

References

Gentzkow, M. and J. Shapiro (2010), *What drives media slant? Evidence from U.S. daily newspapers*. Econometrica 78, 35-7. The full dataset is at http://dx.doi.org/10.3886/ICPSR26242.

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

mnlm, pls, we8there, plot.mnlm, summary.mnlm

Examples

```
data(congress109)

## Bivariate sentiment factors (roll-call vote common scores)
fitCS <- mnlm(congress109Counts, congress109Ideology[,6:7], bins=5, penalty=c(4,1/2))

## plot the fit
plot(fitCS, log='xy', boxwex=.2)

## plot the inverse regression reduction
par(mfrow=c(1,2))
plot(fitCS, type="reduction", v=congress109Ideology$repshare, xlab="Republican Vote-Share",
  covar=1, pch=21, bg=c(4,3,2)[congress109Ideology$party], main="1st common score")
plot(fitCS, type="reduction", v=congress109Ideology$repshare, xlab="Republican Vote-Share",
  covar=2, pch=21, bg=c(4,3,2)[congress109Ideology$party], main="2nd common score")

## example usage of the predict method
predict(fitCS, type="reduction", newdata=congress109Counts[c(68,388),])
predict(fitCS, type="response", newdata=congress109Ideology[c(68,388),6:7])[,c(995,997)]

## example usage of summary method
summary(fitCS, y=congress109Ideology$repshare)

## Fit topic model (use lower tol for true convergence)
par(mfrow=c(1,1))
tpx <- topics(congress109Counts, K=10, tol=100)
plot(tpx, group=congress109Ideology$party=="R", col=c(4,2), labels=c("Dem","GOP"))
summary(tpx)
```

---

corr *Sparse Matrix Correlation*

Description

Correlation calculation for a `simple_triplet_matrix` and a `matrix`.

Usage

```
corr(x, y)
```

Arguments

x	A `simple_triplet_matrix` (or a `matrix`, in which case the function returns `cor(x,y)`).
y	A `matrix` with `nrow(y)=nrow(x)`.

Value

An `ncol(x)` by `ncol(y)` matrix containing correlation between x and y.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

cor, sdev, freq, congress109

Examples

```
data(congress109)
r <- corr(congress109Counts, congress109Ideology$repshare)
## 20 terms for Democrats
sort(r[,1])[1:20]
## 20 terms for Republicans
sort(r[,1], decreasing=TRUE)[1:20]
```

---

freq *Frequency Matrix Conversion*

Description

Convert a count matrix to the corresponding frequency matrix.

Usage

```
freq(x, byrow=TRUE)
```

Arguments

x	A matrix or simple_triplet_matrix with count entries.
byrow	An indicator for whether you have observation-rows and category-columns, or vice versa.

Value

A matrix with row (byrow=TRUE) or column (byrow=FALSE) sums of one.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

corr, pls

Examples

```
freq( t(rmultinom(10, 20, c(1/2,1/4,1/8,1/8))) )
```

`mnlm` *Estimation for high-dimensional Multinomial Logistic Regression*

Description

MAP estimation of multinomial logistic regression models.

Usage

```
mnlm(counts, covars, normalize=TRUE, penalty=c(shape=1,rate=1/2),
                     start=NULL, tol=1e-2, bins=0, verb=FALSE, quasinewton=0, ...)
1
```

Arguments

`counts`	A matrix of multinomial response counts in `ncol(counts)` or `nlevel(counts)` categories for `nrow(counts)` observations. This can be a `matrix`, a vector of response factors, or a `simple_triplet_matrix` (as defined in the `slam` package). Refer to the details for a model identification note.
`covars`	A `matrix` or `simple_triplet_matrix` of `ncol(covars)` covariate values for each of the `nrow(counts)` observations. This does not include the intercept, which is ALWAYS added in the design matrix.
`normalize`	Whether or not to normalize the covariates. Default is `TRUE`. If `covars` is a `matrix`, this will scale the inputs to have mean zero and standard deviation of one. If `covars` is a `simple_triplet_matrix`, we assume that you want to stay in sparse format; hence the inputs are scaled to have sd = 1 but left unshifted.
`penalty`	This input argument is a vector of length 2 containing $[s, r]$ – shape "s" and rate "r" – parameters for the Gamma prior on L1 (lasso) penalty λ, such that $E\lambda = s/r$. Refer to the details section for additional information on this gamma-lasso specification. The default is appropriate for normalized covariates. Additionally, you can specify a normal (ridge) prior with variance 1/`rate` by setting the shape to zero (i.e. with `penalty=c(0,rate)`), set `penalty` to a single fixed value of $\lambda > 0$, or fix coefficients at `start` by giving a penalty of -1. Finally, `penalty` can also be defined as a list with elements containing unique specification for each column of the design matrix (including the intercept).
`start`	An optional initial guess for the full `ncol(covars)+1` by `ncol(counts)` matrix of regression coefficients (including the intercept). Under the default `start=NULL`, the intercept is a logit transform of mean phrase frequencies and coefficients are the correlation between `covars` and `freq(counts)`.
`tol`	Optimization convergence tolerance for the improvement on the un-normalized negative log posterior over a single full parameter sweep.

bins	For faster inference on large data sets (or just to collapse observations across levels for factor covariates), you can specify the number of bins for step-function approximations to the columns of covars. Counts are then collapsed across levels of the interaction between columns of the resulting discrete covariate matrix, typically resulting in a smaller number of observations for estimation.
verb	Control for print-statement output. TRUE prints some initial info and updates every iteration.
quasinewton	If greater than zero, we attempt quasi-Newton acceleration [see Lange, 2010] after the objective updates are less than quasinewton*tol. Be warned: this feature is new and experimental. It can significantly speed convergence, but also increases the chance of a non-global solution.
...	Additional undocumented arguments to internal functions.

Details

Finds the posterior mode for multinomial logistic regression parameters using cyclic coordinate descent. This is designed to be useful for inverse regression analysis of sentiment in text, where the multinomial response is quite large, but should be useful for any large-scale logistic regression.

For binomial response, the first category is assumed null. For multinomial response, the model is identified by placing a Normal(0,1) prior on the intercepts (this can be changed via the list specification for penalty).

Coefficient penalization is based upon the precision parameters λ of independent Laplace priors on each non-intercept regression coefficient. Here, the Laplace density is $p(z) = (\lambda/2)exp[-\lambda|z|]$, with variance $2/\lambda$. Via the penalty argument, this precision is either fixed, which corresponds to the L1 penalty $\lambda|z|$, or it is assigned a $Gamma(s, r)$ prior and estimated jointly with the coefficient, which corresponds to the 'gamma-lasso' non-convex penalty $s * log[1 + |z|/r]$.

In the case of gamma-lasso estimation, prior variance $s/r^2 = E\lambda/r$ controls the degree of penalty curvature. In the case that the variance is large relative to the amount of information in the likelihood, the posterior can become multimodal. Since this leads to unstable optimization and less meaningful MAP estimates, mnlm will warn and automatically double r and s until obtaining a concave posterior. If the resulting prior precision is higher than you would like, it may be worth the computational effort to integrate over penalty uncertainty in mean, rather than MAP, estimation; the reglogit package is available for such inference in binomial regression settings.

Additional details are available in Taddy (2012).

Value

An mnlm object list with entries

intercept	The intercept estimates for each phrase (α).
loadings	A simple_triplet_matrix of estimates for coefficients (Φ) on the scale fitted (possibly normalized) covariates.
counts	simple_triplet_matrix form of the counts input matrix
X	If bins>0, the binned counts matrix used for analysis.
covars	The input covariates, possibly normalized.

V	If bins>0, the binned (and possibly normalized) covariate simple_triplet_matrix used for analysis.
penalty	The penalty specification upon convergence.
normalized	The input normalize indicator.
binned	An indicator for whether the observations was binned.
covarMean	If normalize=TRUE, the amount covariates were shifted (original means for matrix covars, 0 for sparse stm covars). Otherwise empty.
covarSD	If normalize=TRUE, the original covariate standard deviations. Otherwise empty.
prior	The penalty prior (gamma hyperparameters, or fixed laplace scale, or normal precision).
fitted	Fitted count expectations. With binomial response, this is a vector of fitted probabilities. For multinomial response, it is a simple triplet matrix if of fitted probabilities ONLY for non-zero count observations (and with empty entries for zero count observations).

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

Lange (2010), *Numerical Analysis for Statisticians*.

See Also

congress109, we8there, plot.mnlm, summary.mnlm, predict.mnlm

Examples

```
### See congress109 and we8there for more real data examples

### Bernoulli simulation; re-run to see sampling variability ###
n <- 100
v <- rnorm(n)
p <- (1+exp(-(v*2)))^{-1}
y <- rbinom(n, size=1, prob=p)

## fit the logistic model
summary( fit <- mnlm(y, v, verb=TRUE) )
par(mfrow=c(1,2))
plot(fit)

## use predict to see fitted probabilities (could also just use fit$fitted)
phat <-  predict(fit, newdata=matrix(v,ncol=1))
plot(p, phat, pch=21, bg=c(2,4)[y+1], xlab="true probability", ylab="fitted probability")
```

```
### Ripley's Cushing Data ###

## see help(Cushings) for data
library(MASS)
data(Cushings)
train <- Cushings[Cushings$Type != "u",]
newdata <- as.matrix(Cushings[Cushings$Type == "u", 1:2])

## fit, summarize, predict, and plot
fit <- mnlm(counts=factor(train$Type), covars=train[,1:2])
summary(fit)
round(coef(fit),2)
predict(fit, newdata)
par(mfrow=c(1,1))
plot(fit)
```

`normalize` *Normalize*

Description

Normalize matrix columns.

Usage

```
normalize(x, m=NULL, s=NULL, undo=FALSE)
```

Arguments

`x`	A matrix.
`m`	Optional column shifts.
`s`	Optional column scalings.
`undo`	If `undo=TRUE` this will undo a previous normalization. Otherwise, just normalize.

Value

Under default, a matrix with mean-zero and variance-one columns. If shift and scale are specified, a matrix with columns shifted by `-m` and divided by `s`. If `undo=TRUE`, and shift and scale are specified, an un-normalized matrix with column means `m` and standard deviations `s`. In the special case where `x` is a `simple_triplet_matrix` and `m=0`, columns are scaled by `s` but left unshifted and the function returns a `simple_triplet_matrix`.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

freq, corr, sdev, pls

Examples

```
normalize( matrix(1:9, ncol=3) )
```

`plot.mnlm` *Multinomial logistic regression Plots*

Description

Plot function for `mnlm` objects, the output of multinomial logistic regression.

Usage

```
## S3 method for class 'mnlm'
plot(x, type=c("response","reduction","roc"), covar=NULL, v=NULL, xlab=NULL, ylab=NULL, col=NULL, .
```

Arguments

`x`	An output object from the `mnlm` function.
`type`	Under `"response"`, plot the fitted count expectations against observed non-zero counts. Under `"reduction"`, plot the sufficient reduction scores `freq(counts)%*%loadings` from inverse regression based on this `mnlm` fit. Under `"roc"`, plot the receiver operating characteristic for classification based on the fitted model [Note: the roc plot only applies for data with `max(counts)==1` and can be slow if `ncol(counts)` is very large].
`covar`	For `type="reduction"`. The covariate direction to plot. Defaults to 1.
`v`	For `type="reduction"`. Optional argument for the fitted reduction to be plotted against (if, e.g., you wish to plot against unnormalized response).
`xlab`	The x-axis label; will be automatically set if NULL.
`ylab`	The y-axis label; will be automatically set if NULL. For binary data, this becomes the legend title.
`col`	The color(s). Usage changes depending on plot type (for roc, it must be a `ncol(predict(x,x$covars))` length vector).
`...`	Additional `plot` arguments

Value

A fabulous plot.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

See Also

mnlm, congress109, we8there

`plot.pls` *pls plot*

Description

Plot function for Partial Least Squares

Usage

```
## S3 method for class 'pls'
plot(x, K=NULL, xlab="response", ylab=NULL, ...)
```

Arguments

`x`	An output object from the `pls` function.
`K`	The number of pls directions to be used. Can be a vector. If `K`, plot fitted values for `1:fit$K` directions.
`xlab`	The x-axis label.
`ylab`	The y-axis label; if null, will be set to 'pls(k) fitted values' for each k.
`...`	Additional `plot` arguments

Details

Plots response versus fitted values for least-squares fit onto the K pls directions.

Value

A fabulous plot.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

See Also

pls, we8there

`plot.topics` *topic plots*

Description

Plot function for Topic Models

Usage

```
## S3 method for class 'topics'
plot(x, type=c("weight","resid"), group=NULL, labels=NULL,
col=NULL, xlab=NULL, ylab=NULL, main=NULL, tpk=NULL, lgd.K=NULL,
cex.lgdc = 1, cex.lgdt = 1, cex.rmar= 1, ... )
```

Arguments

`x`	An output object from the `topics` function.
`type`	If "weight", the default, provide an image plot of document-topic weights. If "resid", just show a simple histogram of standardized residuals for the positive count entries.
`group`	Optional `logical` vector containing membership in some group for each document; this will be used to color the topic-weight shadings. See the `textir` dataset examples, which color by good reviews for the we8there data or by republicans in congress109.
`labels`	Optional length-two `character` vector of labels for the membership specified in `groups`. `labels[1]` corresponds to `group=FALSE` and `labels[2]` to `group=TRUE`.
`col`	If `type="weight"`, a number from `1:4` specifying the shade color (grey, followed by red, green blue). If `group` is specified, `col[1]` corresponds to `group=FALSE`, and `col[2]` to `group=TRUE`. If `type="resid"`, this is just standard R coloring for the histogram bars.
`xlab`	Optional x-axis label.
`ylab`	Optional y-axis label.
`main`	Optional title.
`tpk`	Optional list of topics to plot. Defaults to `1:x$K`.
`lgd.K`	Optional number of topic-increments (along the X-axis) outside of the plot region at which the legend is centered.
`cex.lgdc`	Magnification factor for legend color-boxes.
`cex.lgdt`	Magnification factor for legend text.
`cex.rmar`	Magnification factor for the right plot margin.
`...`	Additional arguments to the `image` function.

Value

A fabulous plot.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

topics, summary.topics, we8there, congress109, wsjibm

---

pls *Partial Least Squares*

Description

A simple partial least squares procedure.

Usage

```
pls(X, y, K=1, scale=TRUE, verb=TRUE)
```

Arguments

X	The covariate matrix, in either simple_triplet_matrix or matrix format.
y	The response vector.
K	The number of desired PLS directions.
scale	An indicator for whether to standardize X; usually a good idea. If scale=TRUE, X will be scaled to have variance-one columns.
verb	Whether or not to print a small progress script.

Details

Fits the Partial Least Squares algorithm described in Taddy (2011; Section 3.1). In particular, we obtain loadings loadings[,k] as the correlation between X and factors factors[,k], where factors[,1] is initialized at normalize(as.numeric(y)) and subsequent factors are orthogonal to the k'th pls direction, directions[,k]=X%*%loadings[,k].

Value

A pls object list with the following entries

`y`	The response vector.
`X`	The covariate matrix. If `scale=TRUE`, scaled to have variance-one columns.
`directions`	The pls directions `X%*%loadings`.
`factors`	Response factors.
`phi`	The pls loadings.
`fitted`	`K` columns of fitted `y` values for each number of directions.
`fwdmod`	The `lm` object from forward regression `lm(as.numeric(y)~directions)`.
`scale`	If `scale=TRUE` on input, the standard deviations used to scale `X`.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

Wold, H. (1975), *Soft modeling by latent variables: The nonlinear iterative partial least squares approach*. In Perspectives in Probability and Statistics, Papers in Honour of M.S. Bartlett.

See Also

plot.pls, normalize, freq, corr, we8there, congress109

Examples

```
data(congress109)
summary( fit <- pls(freq(congress109Counts), congress109Ideology$repshare, K=3) )
plot(fit, pch=21, bg=c(4,3,2)[congress109Ideology$party])
predict(fit, newdata=freq(congress109Counts[c(68,388),]))

data(we8there)
summary( fit <- pls(freq(we8thereCounts),as.factor(we8thereRatings$Overall)) )
plot(fit, col=c(2,2,2,3,3))
```

`polynomial roots` *Cubic and quadratic function solvers*

Description

Find analytical roots to qubic and quadratic polynomials.

Usage

```
quadratic(b, c, quiet=FALSE, plot=FALSE)
cubic(a, b, c, quiet=FALSE, plot=FALSE)
```

Arguments

`a,b,c`	Polynomial function coefficients (MONIC FORM).
`quiet`	If false, the solution is printed to screen.
`plot`	If true, the function and real root(s) are plotted.

Details

Finds roots to the cubic function $y = x^3 + ax^2 + bx + c$ or quadratic function $y = x^2 + bx + c$.

Value

A list with entries for the coefficients, roots, and solution characterization. In particular,

`type`	The solution characterization: number of complex and real roots.
`coef`	The input coefficients.
`roots`	A vector of the equation roots. For the quadratic equation, if there are complex roots, `roots[1]` is the real part and `roots[2]` is the imaginary part (i.e., complex roots are `roots[1] +- roots[2]*i`). Otherwise, `roots` are (possibly identical) real roots. For the cubic equation, the first root `roots[1]` is always real. If there are complex roots, `roots[2]` is the real part and `roots[3]` is the imaginary part (i.e., complex roots are `roots[2] +- roots[3]*i`). Otherwise, `roots[2:3]` are (possibly identical) real roots.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Abramovitz and Stegun, Handbook of Mathematical Functions, 1972.

See Also

'polyroot' for numerical solutions.

Examples

```
quadratic(1,-2, plot=TRUE)
cubic(0,-15,-4, plot=TRUE)
```

`predict.mnlm` *mnlm predict*

Description

Predict function for Multinomial Logistic Regression

Usage

```
## S3 method for class 'mnlm'
predict( object, newdata, type=c("response","reduction"), ... )
```

Arguments

`object`	An output object from the `mnlm` function.
`type`	Under "reduction", provide the fitted reduction $F\phi$. Under "response", provide the fitted multinomial probabilities.
`newdata`	Under `"response"`, an `ncol(object$loadings)`-column matrix of new covariates. Under `"reduction"`, an `nrow(object$loadings)`-column matrix of multinomial phrase/category counts for new documents/observations. Can be either a simple `matrix` or a `simple_triplet_matrix`.
`...`	Additional unused arguments.

Details

Under `'response'`, this returns fitted multinomial probabilities given new covariate vectors. Under `'reduction'`, we provide the sufficient reduction $F\Phi$ for new documents, with F a document-term frequency matrix (i.e., the counts divided by document totals).

Value

Under `type="response"`, output is an `nrow(newcounts)` by `nrow(object$loadings)` matrix of predicted probabilities for each response category. Under `type="reduction"`, output is an `nrow(newcounts)` by `ncol(object$loadings)` matrix of document scores in each factor (`object$covars`) direction.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

See Also

mnlm, congress109

Examples

```
## fit a congress 109 mnlm model using a random 300 members
data(congress109)
train <- sample(1:529, 300)
counts <- congress109Counts[,col_sums(congress109Counts[train,])>0]
fitRep <- mnlm(counts[train,], congress109Ideology$repshare[train], normalize=TRUE, bins=10)

## extract the reduced dimension text score
Z <- predict(fitRep, newdata=counts[train,], type="reduction")

## use this to build a forward regression model
fwdRep <- lm(repshare ~ Z, data=data.frame(repshare=congress109Ideology$repshare[train], Z=Z[,1]) )

## predict scores for the out-of-sample members
Znew <- predict(fitRep, newdata=counts[-train,], type="reduction")
predicted <- predict(fwdRep, newdata=data.frame(Z=Znew[,1]))
plot(congress109Ideology$repshare[-train], predicted,
     pch=21, bg=c(4,3,2)[congress109Ideology$party[-train]], xlab="repshare")
abline(a=0,b=1, col=8)
```

`predict.pls` *pls predict*

Description

Predict function for Partial Least Squares

Usage

```
## S3 method for class 'pls'
predict( object, newdata, type="response", ... )
```

Arguments

`object`	An output object from the `pls` function.
`newdata`	An `nrow(object$loadings)`-column matrix of multinomial phrase/category counts for new documents/observations. Can be either a simple `matrix` or a `simple_triplet_matrix`.
`type`	If "response", predictions scaled to the original response. If "reduction", fitted partial least squares directions.
`...`	Additional unused arguments.

Details

This function returns the pls projection $X\Phi$ for new covariates, or $\alpha + \beta * X\phi$ if `type="response"` with regression coefficients taken from `object$fwdmod`.

Value

Output is either a vector of predicted resonse or an `nrow(newcounts)` by `ncol(object$loadings)` matrix of pls directions for each new observation.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

See Also

pls, congress109

`predict.topics` *topic predict*

Description

Predict function for Topic Models

Usage

```
## S3 method for class 'topics'
predict( object, newcounts, loglhd=FALSE, ... )
```

Arguments

`object`	An output object from the `topics` function, or the corresponding simple matrix of estimated topics.
`newcounts`	An `nrow(object$theta)`-column matrix of multinomial phrase/category counts for new documents/observations. Can be either a simple `matrix` or a `simple_triplet_matrix`.
`loglhd`	Whether or not to calculate and return `sum(x*log(p))`, the un-normalized log likelihood.
`...`	Additional arguments to the undocumented internal `tpx*` functions.

Details

Under the default mixed-membership topic model, this function uses sequential quadratic programming to fit topic weights Ω for new documents. Estimates for each new ω_i are, conditional on `object$theta`, MAP in the (K-1)-dimensional logit transformed parameter space.

Value

The output is an `nrow(newcounts)` by `object$K` matrix of document topic weights, or a list with including these weights as `W` and the log likelihood as `L`.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

topics, plot.topics, summary.topics, we8there, congress109, wsjibm

Examples

```
## Simulate some data
omega <- t(rdir(500, rep(1/10,10)))
theta <- rdir(10, rep(1/1000,1000))
Q <- omega%*%t(theta)
counts <- matrix(ncol=1000, nrow=500)
totals <- rpois(500, 200)
for(i in 1:500){ counts[i,] <- rmultinom(1, size=totals[i], prob=Q[i,]) }

## predict omega given theta
W <- predict.topics( theta, counts )
plot(W, omega, pch=21, bg=8)
```

rdir *Dirichlet RNG*

Description

Generate random draws from a Dirichlet distribution

Usage

```
rdir(n, alpha)
```

Arguments

n	The number of observations.
alpha	A vector of scale parameters, such that $E[p_j] = \alpha_j / \sum_i \alpha_i$.

Value

An n column matrix containing the observations.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

topics

Examples

```
rdir(3,rep(1,6))
```

sdev *Sparse Matrix Standard Deviation*

Description

Standard deviation for columns of a simple_triplet_matrix.

Usage

```
sdev(x)
```

Arguments

x	A simple_triplet_matrix (or a matrix, in which case the function returns apply(x,2,sd)).

Value

An ncol(x)-length vector containing standard deviations of the columns of x.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

sd, freq, corr, congress109

Examples

```
data(congress109)
sdev(congress109Counts)[1:20]
```

---

summary.mnlm *mnlm summary*

Description

Summary function for Multinomial Logistic Regression

Usage

```
## S3 method for class 'mnlm'
summary( object, y=NULL, ... )
```

Arguments

object	An output object from the mnlm function.
y	A possible response (sentiment) variable of interest in an inverse regression setting.
...	Additional unused arguments.

Details

A short summary function for mnlm objects.

Value

A printout describing the regression coefficients (dimension and sparsity) along with some within-sample correlations or error rates, depending on response and covariate formats. If y is specified and codencolobject$counts>2, we also print the goodness-of-fit R2 for least-squares linear regression of y onto the sufficient reduction `freq(X)%*%loadings`.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

See Also

mnlm, congress109, we8there

`summary.pls` *pls summary*

Description

Summary function for Partial Least Squares

Usage

```
## S3 method for class 'pls'
summary( object, ... )
```

Arguments

`object`	An output object from the `pls` function.
`...`	Additional unused arguments.

Details

A short summary function for pls objects.

Value

A printout of the number of pls directions and the input dimension, followed by a summary of the corresponding forward regression `lm(as.numeric(y)~directions)`.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

See Also

pls

`summary.topics` *topic summary*

Description

Summary function for Topic Models

Usage

```
## S3 method for class 'topics'
summary( object, nwrd=5, tpk=NULL, verb=TRUE, ... )
```

Arguments

`object`	An output object from the `topics` function.
`nwrd`	The number of phrases to output for each topic.
`tpk`	Optional list of topics to summarize. Defaults to `1:x$K`.
`verb`	Whether or not to print the summary.
`...`	Unused arguments from other functions, for S3 compatibility.

Details

This summary orders phrases for each topic according to the lift θ_{kj}/q_j, where q_j is the null-model probability estimate $\sum_i x_{ij} / \sum_i m_i$. This ordering of term relevance can be used to identify representative vocabulary for each topic.

Value

The function prints available model selection results (log Bayes factors, fitted dispersion, and p-value from a test for dispersion > 1) along with usage percentages (i.e. `colMeans(omega)`) and the top `nwrd` phrases by term-lift for each topic in `tpk`. The matrix of top `nwrd` phrases and their associated lift is returned invisibly.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

topics, plot.topics, we8there, congress109, wsjibm

`tfidf` *Term Frequency * Inverse Document Frequency*

Description

Convert a count matrix to the corresponding tfidf matrix.

Usage

```
tfidf(x, freq=FALSE)
```

Arguments

`x`	A matrix or `simple_triplet_matrix`.
`freq`	An indicator for whether `x` is already a frequency matrix.

Value

A matrix with entries $f_{ij} \log[n/d_j]$, where f_{ij} is term-j frequency in document-i, and d_j is the number of documents containing term-j.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

See Also

freq

Examples

```
## 20 important terms
data(congress109)
sort(sdev(tfidf(congress109Counts)), decreasing=TRUE)[1:20]
```

`topics` *Estimation for Topic Models*

Description

MAP estimation of Topic models

Usage

```
topics(counts, K, shape=NULL, initopics=NULL, tol=0.1, bf=FALSE, kill=2, ord=TRUE, verb=1, ...)
```

Arguments

`counts`	A matrix of multinomial response counts in `ncol(counts)` phrases/categories for `nrow(counts)` documents/observations. Can be either a simple `matrix` or a `simple_triplet_matrix`.
`K`	The number of latent topics. If `length(K)>1`, `topics` will find the Bayes factor (vs a null single topic model) for each element and return parameter estimates for the highest probability K.
`shape`	Optional argument to specify the Dirichlet prior concentration parameter as `shape` for topic-phrase probabilities. Defaults to `1/(K*ncol(counts))`. For fixed single `K`, this can also be a `ncol(counts)` by `K` matrix of unique shapes for each topic element.
`initopics`	Optional start-location for $[\theta_1 ... \theta_K]$, the topic-phrase probabilities. Dimensions must accord with the smallest element of `K`. If `NULL`, the initial estimates are built by incrementally adding topics.
`tol`	Convergence tolerance: optimization stops, conditional on some extra checks, when the posterior increase over a full paramater set update is less than `tol`.
`bf`	An indicator for whether or not to calculate the Bayes factor for univariate `K`. If `length(K)>1`, this is ignored and Bayes factors are always calculated.
`kill`	For choosing from multiple `K` numbers of topics (evaluated in increasing order), the search will stop after `kill` consecutive drops in the corresponding Bayes factor. Specify `kill=0` if you want Bayes factors for all elements of `K`.
`ord`	If `TRUE`, the returned topics (columns of `theta`) will be ordered by decreasing usage (i.e., by decreasing `colSums(omega)`).
`verb`	A switch for controlling printed output. `verb > 0` will print something, with the level of detail increasing with `verb`.
`...`	Additional arguments to the undocumented internal `tpx*` functions.

Details

A latent topic model represents each i'th document's term-count vector X_i (with $\sum_j x_{ij} = m_i$ total phrase count) as having been drawn from a mixture of `K` multinomials, each parameterized by topic-phrase probabilities θ_i, such that

$$X_i \sim MN(m_i, \omega_1\theta_1 + ... + \omega_K\theta_K).$$

We assign a K-dimensional Dirichlet(1/K) prior to each document's topic weights $[\omega_{i1}...\omega_{iK}]$, and the prior on each θ_k is Dirichlet with concentration α. The `topics` function uses quasi-newton accelerated EM, augmented with sequential quadratic programming for conditional $\Omega|\Theta$ updates, to obtain MAP estimates for the topic model parameters. We also provide Bayes factor estimation, from marginal likelihood calculations based on a Laplace approximation around the converged MAP parameter estimates. If input `length(K)>1`, these Bayes factors are used for model selection. Full details are in Taddy (2011).

Value

An `topics` object list with entries

`K`	The number of latent topics estimated. If input `length(K)>1`, on output this is a single value corresponding to the model with the highest Bayes factor.
`theta`	The `ncol{counts}` by `K` matrix of estimated topic-phrase probabilities.
`omega`	The `nrow{counts}` by `K` matrix of estimated document-topic weights.
`BF`	The log Bayes factor for each number of topics in the input `K`, against a null single topic model.
`D`	Residual dispersion: for each element of `K`, estimated dispersion parameter (which should be near one for the multinomial), degrees of freedom, and p-value for a test of whether the true dispersion is > 1.
`X`	The input count matrix, in `simple_triplet_matrix` format.

Note

Estimates are actually functions of the MAP (K-1 or p-1)-dimensional logit transformed natural exponential family parameters.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

plot.topics, summary.topics, predict.topics, wsjibm, congress109, we8there

Examples

```
## see wsjibm, congress109, and we8there for data examples

## Simulation Parameters
K <- 10
n <- 100
p <- 100
omega <- t(rdir(n, rep(1/K,K)))
```

```
theta <- rdir(K, rep(1/p,p))

## Simulated counts
Q <- omega%*%t(theta)
counts <- matrix(ncol=p, nrow=n)
totals <- rpois(n, 100)
for(i in 1:n){ counts[i,] <- rmultinom(1, size=totals[i], prob=Q[i,]) }

## Bayes Factor model selection (should choose K or nearby)
summary(simselect <- topics(counts, K=K+c(-5:5)), nwrd=0)

## MAP fit for given K
summary( simfit <- topics(counts,  K=K, verb=2), n=0 )

## Adjust for label switching and plot the fit (color by topic)
toplab <- rep(0,K)
for(k in 1:K){ toplab[k] <- which.min(colSums(abs(simfit$theta-theta[,k]))) }
par(mfrow=c(1,2))
tpxcols <- matrix(rainbow(K), ncol=ncol(theta), byrow=TRUE)
plot(theta,simfit$theta[,toplab], ylab="fitted values", pch=21, bg=tpxcols)
plot(omega,simfit$omega[,toplab], ylab="fitted values", pch=21, bg=tpxcols)
title("True vs Fitted Values (color by topic)", outer=TRUE, line=-2)

## The S3 method plot functions
par(mfrow=c(1,2))
plot(simfit, lgd.K=2)
plot(simfit, type="resid")
```

---

topicVar *topic variance*

Description

Tools for looking at the variance of document-topic weights.

Usage

```
topicVar(counts, theta, omega)
logit(prob)
expit(eta)
```

Arguments

counts	A matrix of multinomial response counts, as inputed to the topics or predict.topics functions.
theta	A fitted topic matrix, as ouput from the topics or predict.topics functions.
omega	A fitted document topic-weight matrix, as ouput from the topics or predict.topics functions.

prob — A probability vector (positive and sums to one) or a matrix with probability vector rows.

eta — A vector of the natural exponential family parameterization for a probability vector (with first category taken as null) or a matrix with each row the NEF parameters for a single observation.

Details

These function use the natural exponential family (NEF) parametrization of a probability vector $q_0...q_{K-1}$ with the first element corresponding to a 'null' category; that is, with $NEF(q) = e_1...e_{K-1}$ and setting $e_0 = 0$, the probabilities are

$$q_k = \frac{exp[e_k]}{1 + \sum exp[e_j]}.$$

Refer to Taddy (2011) for details.

Value

`topicVar` returns an array with dimensions $(K-1, K-1, n)$, where `K=ncol(omega)=ncol(theta)` and `n = nrow(counts) = nrow(omega)`, filled with the posterior covariance matrix for the NEF parametrization of each row of `omega`. Utility `logit` performs the NEF transformation and `expit` reverses it.

Author(s)

Matt Taddy <taddy@chicagobooth.edu>

References

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

topics, predict.topics

---

we8there *On-Line Restaurant Reviews*

Description

Counts for 2804 bigrams in 6175 restaurant reviews from the site www.we8there.com.

Details

The short user-submitted reviews are accompanied by a five-star rating on four specific aspects of restaurant quality - food, service, value, and atmosphere - as well as the overall experience. The reviews originally appear in Maua and Cozman (2009), and the parsing details behind these specific counts are in Taddy (2011).

Value

`we8thereCounts` A `simple_triplet_matrix` of phrase counts indexed by review-rows and bigram-columns.

`we8thereRatings`
A `matrix` containing the associated review ratings.

Author(s)

Matt Taddy, <taddy@chicagobooth.edu>

References

Maua, D.D. and Cozman, F.G. (2009), *Representing and classifying user reviews*. In ENIA '09: VIII Enconro Nacional de Inteligencia Artificial, Brazil.

Taddy (2012), *Multinomial Inverse Regression for Text Analysis*. http://arxiv.org/abs/1012.2098

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

pls, mnlm, congress109

Examples

```
data(we8there)

## use bins to estimate with counts collapsed across equal ratings 1...5
summary( fitwe8 <- mnlm(we8thereCounts, we8thereRatings$Overall, bins=5) )
plot(fitwe8, type="reduction", v=as.factor(we8thereRatings$Overall), col=c(2,2,2,3,3))

## Fit a topic model (use lower tol for true convergence)
tpx <- topics(we8thereCounts, K=10, tol=100)
plot(tpx, group=we8thereRatings$Overall>3, col=c(2,3), labels=c("Bad","Good"))
summary(tpx)
```

---

wsjibm *WSJ Stories on IBM*

Description

Word counts for Wall Street Journal story abstracts with IBM in the title, along with the concurrent returns on IBM stock.

Details

Headlines and one-sentence abstracts for Wall Street Journal (WSJ) stories with IBM in the headline, dating from August 1988 to August 2010, were retrieved from the ProQuest database. Each article is accompanied by two-day return and return-over-market for shares in IBM listed on the New York Stock Exchange, calculated from the opening of the previous day to market close on the day of publication. Full details are available in Taddy (2011).

Value

`wsjibmCounts`	A `simple_triplet_matrix` of counts indexed by article-rows and word-columns.
`wsjibmReturns`	A `matrix` containing the corresponding publication `DATE` along with IBM's two-day holding returns (`RET`) and return over the S&P500 (`ROM`).

Author(s)

Matt Taddy, <taddy@chicagobooth.edu>

References

Taddy (2012), *On Estimation and Selection for Topic Models*. http://arxiv.org/abs/1109.4518

See Also

topics, plot.topics

Examples

```
data(wsjibm)
## fit a simple topic model
summary( newstpx <- topics(wsjibmCounts, K=10, tol=100), nwrd=10 )

## Not run:
## fit topics over years, using prior shape to allow them to change in time
year <- factor(1900 + as.POSIXlt(wsjibmReturns$DATE)$year)
Y <- nlevels(year)
annualtopics <- vector(length=Y, mode="list")
topwords <- c()
shape=NULL
for(i in 1:Y){
      annualtopics[[i]] <- topics(wsjibmCounts[year==levels(year)[i],], K=5, shape=shape, ord=FALSE)
      topwords <- cbind(topwords, as.character(summary(annualtopics[[i]], verb=FALSE)$phrase))
      delta <- 10000 # weight of the previous year in number of words observed per topic
      shape <- annualtopics[[i]]$theta*delta }
## top 5 words by topic in past 4 years
dimnames(topwords) <- list(topic=rep(1:5,each=5), year=levels(year))
print(topwords[,Y - 3:0])
## End(Not run)
```

Index