

# Package ‘maptpx’

July 22, 2025

**Title** MAP Estimation of Topic Models

**Version** 1.9-7

**Author** Matt Taddy <mataddy@gmail.com>

**Depends** R (>= 2.10), slam

**Suggests** MASS

**Description** Maximum a posteriori (MAP) estimation for topic models (i.e., Latent Dirichlet Allocation) in text analysis, as described in Taddy (2012) 'On estimation and selection for topic models'. Previous versions of this code were included as part of the 'textir' package. If you want to take advantage of openmp parallelization, uncomment the relevant flags in src/MAKEVARS before compiling.

**Maintainer** Matt Taddy <mataddy@gmail.com>

**License** GPL-3

**URL** <http://taddylab.com>

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-05-28 10:50:06 UTC

## Contents

counts . . . . .	2
predict.topics . . . . .	2
rdir . . . . .	4
topics . . . . .	4
topicVar . . . . .	7
<b>Index</b>	<b>9</b>

counts

*Utilities for count matrices***Description**

Tools for manipulating (sparse) count matrices.

**Usage**

```
normalize(x,byrow=TRUE)
stm_tfidf(x)
```

**Arguments**

**x** A `simple_triplet_matrix` or `matrix` of counts.  
**byrow** Whether to normalize by row or column totals.

**Value**

`normalize` divides the counts by row or column totals, and `stm_tfidf` returns a matrix with entries  $x_{ij} \log[n/(d_j+1)]$ , where  $x_{ij}$  is term-j frequency in document-i, and  $d_j$  is the number of documents containing term-j.

**Author(s)**

Matt Taddy <mataddy@gmail.com>

**Examples**

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

(x <- matrix(rbinom(15,size=2,prob=.25),ncol=3))
stm_tfidf(x)
```

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 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 $\sum(x \cdot \log(p))$ , 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  $\Omega$  for new documents. Estimates for each new  $\omega_i$  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 <mataddy@gmail.com>

**References**

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

**See Also**

topics, plot.topics, summary.topics, congress109

**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 <mataddy@gmail.com>

**Examples**

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

---

**topics***Estimation for Topic Models*

---

**Description**

MAP estimation of Topic models

**Usage**

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

**Arguments**

counts	A matrix of multinomial response counts in <code>ncol(counts)</code> phrases/categories for <code>nrow(counts)</code> documents/observations. Can be either a simple matrix or a <code>simple_triplet_matrix</code> .
K	The number of latent topics. If <code>length(K)&gt;1</code> , <code>topics</code> 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 $1/(K*\text{ncol}(\text{counts}))$ . For fixed single K, this can also be a <code>ncol(counts)</code> by K matrix of unique shapes for each topic element.
inittopics	Optional start-location for $[\theta_1 \dots \theta_K]$ , 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 <i>absolute</i> posterior increase over a full parameter set update is less than <code>tol</code> .
bf	An indicator for whether or not to calculate the Bayes factor for univariate K. If <code>length(K)&gt;1</code> , 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 <code>kill</code> consecutive drops in the corresponding Bayes factor. Specify <code>kill=0</code> if you want Bayes factors for all elements of K.
ord	If TRUE, the returned topics (columns of <code>theta</code> ) will be ordered by decreasing usage (i.e., by decreasing <code>colSums(omega)</code> ).
verb	A switch for controlling printed output. <code>verb &gt; 0</code> will print something, with the level of detail increasing with <code>verb</code> .
...	Additional arguments to the undocumented internal <code>tpx*</code> 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  $\theta_i$ , such that

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

We assign a K-dimensional Dirichlet( $1/K$ ) prior to each document's topic weights  $[\omega_{i1} \dots \omega_{iK}]$ , and the prior on each  $\theta_k$  is Dirichlet with concentration  $\alpha$ . 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 <code>length(K)&gt;1</code> , on output this is a single value corresponding to the model with the highest Bayes factor.
theta	The <code>ncol{counts}</code> by K matrix of estimated topic-phrse probabilities.
omega	The <code>nrow{counts}</code> 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 <code>dgTMatrix</code> 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 <mataddy@gmail.com>

**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**

```
## 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)), nwr=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	<i>topic variance</i>
----------	-----------------------

---

## 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 <code>topics</code> or <code>predict.topics</code> functions.
theta	A fitted topic matrix, as output from the <code>topics</code> or <code>predict.topics</code> functions.
omega	A fitted document topic-weight matrix, as output from the <code>topics</code> or <code>predict.topics</code> 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  $q_0 \dots q_{K-1}$  with the first element corresponding to a 'null' category; that is, with  $NEF(q) = e_1 \dots e_{K-1}$  and setting  $e_0 = 0$ , the probabilities are

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

Refer to Taddy (2012) for details.

**Value**

topicVar returns an array with dimensions  $(K-1, K-1, n)$ , where  $K = \text{ncol}(\text{omega}) = \text{ncol}(\text{theta})$  and  $n = \text{nrow}(\text{counts}) = \text{nrow}(\text{omega})$ , 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 <mataddy@gmail.com>

**References**

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

**See Also**

topics, predict.topics



# Index

`counts`, [2](#)

`expit(topicVar)`, [7](#)

`logit(topicVar)`, [7](#)

`normalize(counts)`, [2](#)

`predict.topics`, [2](#)

`rdir`, [4](#)

`stm_tfidf(counts)`, [2](#)

`topics`, [4](#)

`topicVar`, [7](#)