

# Package ‘mlstm’

April 3, 2026

**Type** Package

**Title** Multilevel Supervised Topic Models with Multiple Outcomes

**Version** 0.1.6

**Description** Fits latent Dirichlet allocation (LDA), supervised topic models, and multilevel supervised topic models for text data with multiple outcome variables. Core estimation routines are implemented in C++ using the 'Rcpp' ecosystem.

For topic models, see Blei et al. (2003) <<https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>>.

For supervised topic models, see Blei and McAuliffe (2007) <[https://papers.nips.cc/paper\\_files/paper/2007/hash/d56b9fc4b0f1be8871f5e1c40c0067e7-Abstract.html](https://papers.nips.cc/paper_files/paper/2007/hash/d56b9fc4b0f1be8871f5e1c40c0067e7-Abstract.html)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** R (>= 4.0.0)

**Imports** Rcpp, Matrix, data.table, RcppParallel, stats

**LinkingTo** Rcpp, RcppArmadillo, RcppParallel, BH

**SystemRequirements** C++17

**RoxygenNote** 7.3.3

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://thimeno1993.github.io/mlstm/>

**BugReports** <https://github.com/thimeno1993/mlstm/issues>

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Tomoya Himeno [aut, cre]

**Maintainer** Tomoya Himeno <[bd24f002@eg.hit-u.ac.jp](mailto:bd24f002@eg.hit-u.ac.jp)>

**Repository** CRAN

**Date/Publication** 2026-04-03 09:00:10 UTC

## Contents

eLDA_pass_b_fast . . . . .	2
init_mod_from_count . . . . .	3
run_lda_gibbs . . . . .	4
run_mlstm_vi . . . . .	5
run_stm_vi . . . . .	8
set_threads . . . . .	10
stm_multi_hier_vi_parallel . . . . .	11
stm_vi_parallel . . . . .	12

<b>Index</b>	<b>15</b>
--------------	-----------

---

eLDA_pass_b_fast	<i>One Gibbs sampling sweep for LDA (collapsed) using document–term list.</i>
------------------	---

---

## Description

This function performs a single collapsed Gibbs sampling pass over all non-zero document–term entries. Each  $(d, v, \text{count})$  triple is treated as ‘count’ replicated word tokens sharing the same topic assignment.

## Usage

```
eLDA_pass_b_fast(mod, count, ndsum, NZ, V, K, alpha, beta)
```

## Arguments

mod	List with current sampler state: $z$ , $nd$ , $nw$ , and $nwsum$ as described above.
count	IntegerMatrix of size $NZ \times 3$ , where each row is a triple $(d, v, c)$ with 0-based indices: document index $d$ , word index $v$ , and count $c$ for that (doc, word) pair.
ndsum	IntegerVector of length $D$ ; total token count per document (i.e., $ndsum[d] = \sum_k nd(d, k)$ ). Updated in place.
NZ	Integer, number of non-zero entries (rows in count and length of $z$ ).
V	Integer, vocabulary size.
K	Integer, number of topics.
alpha	Scalar Dirichlet prior parameter for document–topic distributions $\theta_d$ (symmetric).
beta	Scalar Dirichlet prior parameter for topic–word distributions $\phi_k$ (symmetric).

**Details**

The state is stored in a list ‘mod‘ containing:

**z** Integer vector of length NZ; topic assignment for each (d, v, count) triple.

**nd** D×K integer matrix; document–topic counts.

**nw** K×V integer matrix; topic–word counts.

**nwsun** Integer vector of length K; total word count per topic.

**Value**

A list with updated state:

**z** Updated topic assignment vector (length NZ).

**nd** Updated D×K document–topic counts.

**nw** Updated K×V topic–word counts.

**nwsun** Updated total word counts per topic.

---

init\_mod\_from\_count    *Initialize LDA/STM state from a (d, v, c) sparse count matrix.*

---

**Description**

Given a document-term matrix in triplet form (d, v, c) using 0-based indices, this function initializes the LDA state: - samples initial topic assignments z, - constructs document-topic counts nd, - constructs topic-word counts nw, - computes ndsum, nwsun, and normalized topic proportions X.

**Usage**

```
init_mod_from_count(count, K = NULL, phi = NULL, seed = NULL)
```

**Arguments**

count	Integer matrix with 3 columns representing triples (d, v, c), where d and v are 0-based indices.
K	Integer, number of topics. Required if ‘phi‘ is NULL. If ‘phi‘ is provided, K is inferred from ncol(phi).
phi	Optional numeric matrix of size V x K specifying per-word topic probabilities used only during initialization.
seed	Optional integer random seed.

**Details**

If a topic-word probability matrix ‘phi‘ is provided (V x K), initial topics are sampled according to phi[v+1, ]. Otherwise, topics are sampled uniformly from K topics.

**Value**

A list with components:

**z** Integer vector (length NZ) of sampled topics, 0-based.

**nd** DxK document-topic count matrix.

**nw** KxV topic-word count matrix.

**ndsum** Integer vector (length D) with row sums of nd.

**nwsun** Integer vector (length K) with row sums of nw.

**X** DxK matrix of normalized topic proportions nd / ndsum.

**D** Number of documents.

**V** Vocabulary size.

**K** Number of topics.

**NZ** Number of non-zero entries (rows in count).

---

run\_lda\_gibbs

*Collapsed LDA Gibbs sampling for sparse (d, v, c) triplet data.*


---

**Description**

This function performs collapsed Gibbs sampling for the standard LDA model using a sparse document-term representation:

1. initializes the LDA state via `init_mod_from_count()`,
2. runs `n_iter` iterations of the C++ Gibbs kernel `eLDA_pass_b_fast()`,
3. returns the final model state, including posterior topic-word and document-topic distributions.

**Usage**

```
run_lda_gibbs(
  count,
  K,
  alpha,
  beta,
  n_iter = 100L,
  phi = NULL,
  seed = NULL,
  verbose = TRUE,
  progress_every = 10L
)
```

**Arguments**

count	Integer matrix of size NZ x 3 with rows (d, v, c) in 0-based indexing: document index d, word index v, and count c for that pair.
K	Integer, number of topics. Required unless phi is supplied. If phi is provided, K is inferred from ncol(phi).
alpha	Scalar Dirichlet prior parameter for document-topic distributions.
beta	Scalar Dirichlet prior parameter for topic-word distributions.
n_iter	Integer, number of Gibbs sweeps to run.
phi	Optional V x K topic-word probability matrix used only for initializing topic assignments in <code>init_mod_from_count()</code> .
seed	Optional integer random seed passed to the initializer.
verbose	Logical; if TRUE, print progress messages.
progress_every	Integer; print progress every this many iterations.

**Value**

A list mod containing:

**z** Integer vector of length NZ; final topic assignments (0-based).

**nd** D x K document-topic count matrix.

**nw** K x V topic-word count matrix.

**ndsum** Integer vector of length D; document token counts.

**nwsun** Integer vector of length K; topic token counts.

**phi** V x K topic-word posterior mean  $p(w | z = k)$  computed from nw.

**theta** D x K document-topic posterior mean  $p(z = k | d)$  computed from nd.

**loglik\_trace** Vector of log-likelihoods.

**D** Number of documents.

**V** Vocabulary size.

**K** Number of topics.

**NZ** Number of non-zero (d, v, c) entries.

---

run_mlstm_vi	<i>Multi-level supervised topic model (MLSTM) via variational inference.</i>
--------------	--

---

**Description**

This function fits a multi-output supervised LDA model with a hierarchical prior on regression coefficients:

$$\eta_j \sim N(\mu, \Lambda^{-1}), \quad \Lambda \sim \text{IW}(v, \Omega).$$

**Usage**

```

run_mlstm_vi(
  count,
  Y,
  K,
  alpha,
  beta,
  mu,
  epsilon,
  Omega,
  phi = NULL,
  seed = NULL,
  max_iter = 200L,
  min_iter = 50L,
  tol_elbo = 1e-04,
  update_sigma = TRUE,
  tau = 20L,
  exact_second_moment = FALSE,
  show_progress = TRUE,
  chunk = 5000L,
  verbose = TRUE,
  sigma2_init = NULL
)

```

**Arguments**

count	Integer matrix with 3 columns (d, v, c), using 0-based indices. Each row represents document index d, word index v, and token count c.
Y	Numeric matrix of size D x J containing J response variables for each of the D documents. NA values are allowed and are ignored in the initial regression used to seed eta and sigma2.
K	Integer, number of topics. Required if phi is NULL; ignored if phi is supplied, in which case K = ncol(phi).
alpha	Dirichlet prior parameter for document-topic distributions.
beta	Dirichlet prior parameter for topic-word distributions.
mu	Numeric vector of length K; prior mean for each $\eta_j$ .
epsilon	Scalar degrees of freedom for the inverse-Wishart prior on the precision matrix $\Lambda$ .
Omega	Numeric K x K positive-definite scale matrix for the inverse-Wishart prior.
phi	Optional numeric matrix of size V x K used only to initialize topic assignments via <code>init_mod_from_count()</code> .
seed	Optional integer random seed used for initialization.
max_iter	Maximum number of variational sweeps.
min_iter	Minimum number of sweeps before checking convergence.

tol_elbo	Numeric tolerance for the relative ELBO change used in the convergence criterion.
update_sigma	Logical; if TRUE, update sigma2 inside stm_multi_hier_vi_parallel(). If FALSE, keep sigma2 fixed at its initialized value.
tau	Log-space cutoff for local topic responsibilities in the C++ routine (controls pruning for stability and speed).
exact_second_moment	Logical; reserved flag intended to control whether the exact second moment $E[\bar{z}\bar{z}^\top]$ is accumulated in the E-step. <b>**Currently this option has no effect**</b> : the underlying C++ implementation ignores the accumulated second-moment matrix when updating the variational parameters, and only an approximate moment based on $\bar{z}\bar{z}^\top$ is effectively used.
show_progress	Logical; forwarded to stm_multi_hier_vi_parallel().
chunk	Integer; number of documents per parallel block in the C++ E-step.
verbose	Logical; if TRUE, print ELBO and its relative change at each sweep.
sigma2_init	Optional numeric scalar or length-J vector specifying the initial noise variances. If NULL, sigma2 is estimated for each response dimension by least squares regression of $Y[, j]$ on initial topic proportions.

## Details

The latent topic layer is standard LDA, and each response dimension  $j$  follows a Gaussian regression on document-level topic proportions. Variational inference is performed by repeated calls to the C++ routine `stm_multi_hier_vi_parallel()` until convergence or a maximum number of sweeps is reached.

Convergence is assessed based on the relative changes in the evidence lower bound (ELBO) and the supervised label log-likelihood:

$$\frac{\text{ELBO}_t - \text{ELBO}_{t-1}}{|\text{ELBO}_{t-1}|}, \quad \frac{\ell_t - \ell_{t-1}}{|\ell_{t-1}|}.$$

After a minimum number of iterations, the algorithm is declared to have converged when both quantities are non-negative and smaller than the prescribed tolerance.

## Value

A list mod containing (at least):

**nd**  $D \times K$  document-topic counts.

**nw**  $K \times V$  topic-word counts.

**ndsum** Integer vector of length  $D$ ; document token counts.

**nwsun** Integer vector of length  $K$ ; topic token counts.

**eta**  $K \times J$  matrix of regression coefficients.

**sigma2** Length- $J$  vector of noise variances.

**Lambda\_E**  $K \times K$  posterior mean of  $\Lambda$  (if returned by C++).

**IW\_***epsilon\_hat* Posterior degrees of freedom (if returned by C++).

**IW\_***Omega\_hat* Posterior scale matrix (if returned by C++).

**phi**  $V \times K$  topic-word posterior mean  $p(w | z = k)$  computed from nw.

**theta**  $D \times K$  document-topic posterior mean  $p(z = k | d)$  computed from nd.

**elbo** Final ELBO value.

**label\_loglik** Final label log-likelihood term.

**elbo\_trace** Numeric vector of ELBO values over iterations.

**label\_loglik\_trace** Numeric vector of label log-likelihoods.

**n\_iter** Number of sweeps actually performed.

**D** Number of documents.

**V** Vocabulary size.

**K** Number of topics.

**J** Number of response dimensions.

**NZ** Number of non-zero (d, v, c) entries.

---

run_stm_vi	<i>Supervised topic model (STM) variational inference with ELBO-based convergence.</i>
------------	--

---

## Description

This function performs supervised topic model (STM) using variational inference. It initializes topic assignments from count (optionally using a topic-word prior phi), estimates regression parameters, and repeatedly calls the C++ routine `stm_vi_parallel()` until convergence.

## Usage

```
run_stm_vi(
  count,
  y,
  K,
  alpha,
  beta,
  phi = NULL,
  seed = NULL,
  max_iter = 200L,
  min_iter = 50L,
  tol_elbo = 1e-04,
  update_sigma = TRUE,
  tau = 20L,
  show_progress = TRUE,
  chunk = 5000L,
  verbose = TRUE,
  sigma2_init = NULL
)
```

**Arguments**

count	Integer matrix with 3 columns (d, v, c) in 0-based indexing. Each row represents document index d, word index v, and token count c.
y	Numeric vector of length D. Must not contain NA values.
K	Integer, number of topics. Required if phi is NULL; ignored if phi is provided (then $K = \text{ncol}(\text{phi})$ ).
alpha	Dirichlet prior parameter for document-topic distributions.
beta	Dirichlet prior parameter for topic-word distributions.
phi	Optional V x K topic-word probability matrix used only for initializing topic assignments.
seed	Optional integer random seed used in the initialization step.
max_iter	Maximum number of variational sweeps.
min_iter	Minimum number of sweeps before checking ELBO convergence.
tol_elbo	Numeric tolerance for relative ELBO change.
update_sigma	Logical; if TRUE, update sigma2 each sweep.
tau	Numeric log-space cutoff used in <code>stm_vi_parallel()</code> .
show_progress	Logical; print low-level progress inside C++.
chunk	Integer; number of documents per parallel block.
verbose	Logical; print ELBO and relative change per sweep.
sigma2_init	Optional numeric scalar specifying the initial noise variance. If NULL, sigma2 is estimated once by least squares.

**Details**

Convergence is assessed based on the relative changes in the evidence lower bound (ELBO) and the supervised label log-likelihood:

$$\frac{\text{ELBO}_t - \text{ELBO}_{t-1}}{|\text{ELBO}_{t-1}|}, \quad \frac{\ell_t - \ell_{t-1}}{|\ell_{t-1}|}.$$

After a minimum number of iterations, the algorithm is declared to have converged when both quantities are non-negative and smaller than the prescribed tolerance.

**\*\*Important:\*\*** This function assumes that the response vector y contains **\*\*no NA\*\*** values. The underlying C++ implementation does not skip missing responses and requires `y[d]` to be finite for all documents.

**Value**

A list containing:

**nd** D x K document-topic count matrix.

**nw** K x V topic-word count matrix.

**ndsum** Length-D vector of document token counts.

**nwsun** Length-K vector of topic token counts.  
**eta** K-dimensional regression coefficient vector.  
**sigma2** Final noise variance.  
**phi** V x K topic-word posterior mean.  
**theta** D x K document-topic posterior mean.  
**elbo** Final ELBO.  
**label\_loglik** Final supervised term.  
**elbo\_trace** ELBO values per sweep.  
**label\_loglik\_trace** Label log-likelihood per sweep.  
**n\_iter** Number of iterations actually performed.  
**D, V, K, NZ** Model dimensions.

---

set\_threads

*Set threading options for STM/MLSTM computations*


---

### Description

This helper configures OpenMP/BLAS threads to ensure reproducible and stable performance across the low-level C++ routines used by the package.

### Usage

```
set_threads(num_threads = NULL)
```

### Arguments

**num\_threads** Integer number of threads. If NULL, use (cores - 1).

### Value

Invisibly returns an integer giving the number of threads used.

---

 stm\_multi\_hier\_vi\_parallel

*Variational inference for multi-output supervised topic models with hierarchical prior.*

---

## Description

The model includes: - LDA structure:  $\theta_{d,j} \sim \text{Dir}(\alpha)$ ,  $\phi_{k,j} \sim \text{Dir}(\beta)$  - Gaussian response:  $y_{[d,j]} \sim \text{N}(\bar{z}_{d,j} \eta_j, \sigma_j^2)$  - Hierarchical prior:  $\eta_j \sim \text{N}(\mu, \Lambda^{-1})$   $\Lambda \sim \text{inverse-Wishart}(\nu, \Omega)$

## Usage

```
stm_multi_hier_vi_parallel(
  mod,
  docs,
  y,
  ndsum,
  NZ,
  V,
  K,
  J,
  alpha,
  beta,
  mu,
  epsilon,
  Omega,
  update_sigma = TRUE,
  tau = 20L,
  exact_second_moment = FALSE,
  show_progress = TRUE,
  chunk = 5000L
)
```

## Arguments

mod	List with model state: - nd (D x K) document-topic counts - nw (K x V) topic-word counts - eta (K x J) regression coefficients - sigma2 (J) noise variances
docs	IntegerMatrix (NZ x 3) with (doc_id, word_id, count).
y	NumericMatrix (D x J) response matrix.
ndsum	IntegerVector (D) document token counts.
NZ, V, K, J	Model dimensions.
alpha, beta	Dirichlet hyperparameters.
mu	NumericVector (K) prior mean.
epsilon	Degrees of freedom for inverse-Wishart.

Omega	Scale matrix for inverse-Wishart.
update_sigma	Logical; update sigma2 or not.
tau	Numeric cutoff for stability.
exact_second_moment	Logical flag (currently not used).
show_progress	Logical; print progress.
chunk	Integer; documents per parallel block.

### Value

A list with updated variational parameters and diagnostics:

**nd** D x K integer matrix of document-topic counts.

**nw** K x V integer matrix of topic-word counts.

**eta** K x J numeric matrix of regression coefficients.

**sigma2** Length-J numeric vector of noise variances.

**Lambda\_E** K x K numeric matrix, posterior mean of precision matrix Lambda.

**IW\_upsilon\_hat** Numeric scalar, posterior degrees of freedom.

**IW\_Omega\_hat** K x K numeric matrix, posterior scale matrix.

**elbo** Numeric scalar, evidence lower bound.

**label\_loglik** Numeric scalar, supervised log-likelihood term.

---

stm_vi_parallel	<i>Variational inference for supervised LDA (single continuous response).</i>
-----------------	---

---

### Description

The model combines unsupervised topic modeling (LDA) with a Gaussian response on document-level topic proportions.

### Usage

```
stm_vi_parallel(
  mod,
  docs,
  y,
  ndsum,
  NZ,
  V,
  K,
  alpha,
  beta,
  update_sigma = TRUE,
```

```

    tau = 20L,
    show_progress = TRUE,
    chunk = 5000L
)

```

## Arguments

mod	A list containing the current model state: <b>nd</b> D x K matrix of document-topic counts. <b>nw</b> K x V matrix of topic-word counts. <b>eta</b> Numeric vector of length K; regression coefficients. <b>sigma2</b> Scalar noise variance for the Gaussian response.
docs	IntegerMatrix of size NZ x 3, where each row is a triple (d, v, c) in 0-based indexing: document index d, word index v, and count c = n_dv. Rows with d outside [0, D-1] are ignored.
y	NumericVector of length D; response y_d for each document.
ndsum	IntegerVector of length D; total token count per document (that is, ndsum[d] = sum_v n_dv).
NZ	Integer, number of non-zero entries in docs (rows of docs).
V	Integer, vocabulary size.
K	Integer, number of topics.
alpha	Scalar Dirichlet prior parameter for document-topic distributions theta_d (symmetric prior with parameter alpha).
beta	Scalar Dirichlet prior parameter for topic-word distributions phi_k (symmetric prior with parameter beta).
update_sigma	Logical; if TRUE, update the noise variance sigma2 from residuals y_d - zbar_d^T eta, otherwise keep sigma2 fixed.
tau	Numeric, log-space cutoff used to prune very small topic responsibilities phi[d,i,k] for numerical stability and efficiency.
show_progress	Logical; if TRUE, print simple progress output during the E-step over documents.
chunk	Integer, number of documents to process per parallel block in the E-step. Larger values reduce overhead but may use more memory.

## Details

$$y_d \sim N(\text{zbar}_d^T \text{eta}, \text{sigma}^2).$$

This function performs one variational inference sweep with a parallel document-level E-step and simple updates for the regression parameters.

**Value**

A list with updated variational parameters and diagnostics:

**nd** Updated  $D \times K$  document-topic counts.

**nw** Updated  $K \times V$  topic-word counts.

**eta** Updated  $K$ -dimensional regression coefficient vector.

**sigma2** Updated scalar noise variance.

**elbo** Scalar evidence lower bound (approximate).

**label\_loglik** Gaussian response log-likelihood component.

# Index

eLDA\_pass\_b\_fast, [2](#)

init\_mod\_from\_count, [3](#)

run\_lda\_gibbs, [4](#)

run\_mlstm\_vi, [5](#)

run\_stm\_vi, [8](#)

set\_threads, [10](#)

stm\_multi\_hier\_vi\_parallel, [11](#)

stm\_vi\_parallel, [12](#)