Package: incidental (via r-universe)

October 22, 2024

Title Implements Empirical Bayes Incidence Curves

Version 0.1

Description Make empirical Bayes incidence curves from reported case data using a specified delay distribution.

Depends R (>= 3.5.0)

License MIT + file LICENSE

LazyData true

RoxygenNote 7.1.1

Imports ggplot2, MASS, matrixStats, numDeriv, dlnm, stats, utils

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Andrew Miller [aut], Lauren Hannah [aut, cre], Nicholas Foti [aut], Joseph Futoma [aut], Apple, Inc. [cph]

Maintainer Lauren Hannah <lauren_hannah@apple.com>

Repository CRAN

Date/Publication 2020-09-16 09:50:03 UTC

Contents

`compute_expected_cases` *Compute expected cases*

Description

This function computes expected cases given incidence curve parameters and a delay distribution.

Usage

```
compute_expected_cases(beta, Q, lnPmat, Tobs)
```

Arguments

`beta`	parameter vector of num_params
`Q`	spline basis matrix, of size Tmod x num_params
`lnPmat`	matrix size Tobs x Tobs, log of make_likelihood_matrix
`Tobs`	maximum observed time point

Value

A Tobs-length vector that models expected cases

compute_log_incidence *Compute log likelihood of incidence model*

Description

This function computes log likelihood of incidence model given parameters and observations.

Usage

```
compute_log_incidence(beta, Q, Tobs)
```

Arguments

beta	parameter vector of num_params
Q	spline basis matrix, of size Tmod x num_params
Tobs	maximum observed time point

Value

I Tobs-length vector that models log incidence curve

covid_delay_dist *Delay distribution from COVID-19 pandemic.*

Description

Daily case, hospitalization, and death proportions.

Usage

```
covid_delay_dist
```

Format

A data frame with 61 entries and 4 columns.

days number of days since infection

case proportion of cases confirmed by a test that are recorded on that day

hospitalization proportion of cases that become hospitalized that are hospitalized on that day

death proportion of cases that result in death that die on that day

Source

Time from incidence to symptoms: Lauer et al., "Estimated Incubation Period of COVID-19", ACC (2020). https://www.acc.org/latest-in-cardiology/journal-scans/2020/05/11/15/18/the-incubation-period-of-coronavirus-disease.

Time from symptoms to recorded cases: Case line data from Florida through 2020-07-14 with same day waits removed. https://open-fdoh.hub.arcgis.com/datasets/florida-covid19-case-line-data.

Time from symptoms to hospitalization: Wang et al., "Clinical Characteristics of 138 Hospitalized Patients With 2019 Novel Coronavirus–Infected Pneumonia in Wuhan, China", JAMA (2020). https://jamanetwork.com/journals/jama/fullarticle/2761044.

Time from hospitalization to death: Lewnard et al. "Incidence, clinical outcomes, and transmission dynamics of severe coronavirus disease 2019 in California and Washington: prospective cohort study", BJM (2020). https://www.bmj.com/content/369/bmj.m1923.long

---

covid_new_york_city *New York City data from the COVID-19 pandemic.*

Description

Daily case, hospitalization, and death proportions by borough through 2020-06-30.

Usage

```
covid_new_york_city
```

Format

A data frame with 615 entries and 5 columns.

date record date

borough record borough: Brooklyn, Bronx, Manhattan, Queens, and Staten Island

case number of recorded cases

hospitalization number of new hospital admissions

death number of recorded deaths

Source

New York City Department of Health https://raw.githubusercontent.com/nychealth/coronavirus-data/master/boro/boroughs-case-hosp-death.csv.

data_check *Input data check*

Description

Check input data for:

- minimum length of reported
- integer for reported
- positivity for delay_dist and reported
- sums to 1 for delay_dist

Throw an error if any conditions are violated.

Usage

```
data_check(reported, delay_dist)
```

Arguments

reported	An integer vector of reported cases.
delay_dist	A positive vector that sums to one, which describes the delay distribution.

data_processing *Data processing wrapper*

Description

Does basic checks for reported data and delay distribution, front pads, and makes AR extrapolation.

Usage

```
data_processing(
  reported,
  delay_dist,
  num_ar_steps = 10,
  num_ar_samps = 100,
  seed = 1,
  linear_tail = 14,
  front_pad_size = 10,
  extrapolation_prior_precision = 2
)
```

Arguments

reported	An integer vector of reported cases.
delay_dist	A positive vector that sums to one, which describes the delay distribution.
num_ar_steps	An integer number of AR steps after last observation.
num_ar_samps	An integer number of AR samples.
seed	Seed for RNG.
linear_tail	An integer number of days used to fit linear model on tail to be used as a mean for AR extrapolation.
front_pad_size	An integer for initial number of 0's before first observation.
extrapolation_prior_precision	A positive scalar for extrapolation slope shrinkage prior precision.

Value

A list with elements:

- extrap = a matrix of size (num_ar_samps x n + num_ar_steps + front_pad_size)
- original = a vector of logicals for whether in original time series range

---

diff_trans *Transpose of the 1st difference operator*

Description

This function computes a transpose of the 1st difference operator.

Usage

```
diff_trans(a)
```

Arguments

a	A vector of inputs

Value

The transpose of the first difference operator

`fit_incidence` *Fit incidence curve to reported data*

Description

This is a function that fits an incidence curve to a set of reported cases and delay distribution using an empirical Bayes estimation method, which fits parameters for a spline basis. All hyper parameter tuning and data processing are done within this function.

Usage

```
fit_incidence(
  reported,
  delay_dist,
  dof_grid = seq(6, 20, 2),
  dof_method = "aic",
  lam_grid = 10^(seq(-1, -8, length.out = 20)),
  lam_method = "val",
  percent_thresh = 2,
  regularization_order = 2,
  num_ar_steps = 10,
  num_ar_samps = 100,
  linear_tail = 14,
  front_pad_size = 10,
  extrapolation_prior_precision = 10,
  frac_train = 0.75,
  fisher_approx_cov = TRUE,
  end_pad_size = 50,
  num_samps_per_ar = 10,
  val_restarts = 2,
  seed = 1
)
```

Arguments

`reported`	An integer vector of reported cases.
`delay_dist`	A positive vector that sums to one, which describes the delay distribution.
`dof_grid`	An integer vector of degrees of freedom for the spline basis.
`dof_method`	Metric to choose "best" spline degrees of freedom: 'aic': Akaike information criterion, 'bic': Bayesian information criterion, 'val': validation likelihood.
`lam_grid`	A vector of regularization strengths to scan.
`lam_method`	metric to choose "best" regularization strength lambda: 'aic': Akaike information criterion, 'bic': Bayesian information criterion, 'val': validation likelihood.
`percent_thresh`	If using validation likelihood to select best, the largest (strongest) lambda that is within 'percent_thresh' of the highest validation lambda will be selected. Default is 2. Must be greater than 0.

`regularization_order`
: An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

`num_ar_steps`
: An integer number of AR steps after last observation.

`num_ar_samps`
: An integer number of AR samples.

`linear_tail`
: An integer number of days used to fit linear model on tail to be used as a mean for AR extrapolation.

`front_pad_size`
: An integer for initial number of 0's before first observation.

`extrapolation_prior_precision`
: A positive scalar for extrapolation slope shrinkage prior precision.

`frac_train`
: A numeric between 0 and 1 for fraction of data used to train lambda validation.

`fisher_approx_cov`
: A flag to use either the Fisher Information (TRUE) or the Hessian (FALSE) to approx posterior covariance over parameters.

`end_pad_size`
: And integer number of steps the spline is defined beyond the final observation.

`num_samps_per_ar`
: An integer for the number of Laplace samples per AR fit.

`val_restarts`
: An integer for the number of times to refit hyperparameters if 'val' is used for either. Set to 1 for faster but more unstable fits.

`seed`
: Seed for RNG.

Value

A list with the following entries:

- Isamps – sample of the incidence curve from a Laplace approximation per AR sample;
- Ihat – MAP incidence curve estimate;
- Chat – expected cases given MAP incidence curve estimate;
- beta_hats – matrix of beta's per AR sample;
- best_dof – best degrees of freedom from tuning;
- best_lambda – best regularization parameter from tuning; and
- reported – a copy of reported values used for fitting.

Examples

```
indiana_model <- fit_incidence(
                 reported = spanish_flu$Indiana,
                 delay_dist = spanish_flu_delay_dist$proportion)
```

front_zero_pad *Pad reported data with zeros in front*

Description

Add zeros in front of reported data avoid infections from before first reported date all being placed on first reported date.

Usage

```
front_zero_pad(reported, size)
```

Arguments

reported	An integer vector of reported cases
size	An integer size of zero-padding

Value

An integer vector of cases with size 0's in front

incidence_to_df *Export incidence model to data frame*

Description

Export the output of fit_incidence to a data frame with an optional addition of a time index.

Usage

```
incidence_to_df(x, times = NULL, low_quantile = 0.05, high_quantile = 0.95)
```

Arguments

x	An "incidence_spline_model" output from fit_incidence.
times	An optional vector of time indices.
low_quantile	A scalar that specifies the low quantile value for the output CI.
high_quantile	A scalar that specifies the high quantile value for the output CI.

Value

A data frame with the following entries:

- Time – a time index; if ‘ts‘ is ‘NULL‘ it is the observation number;
- Reported – the value of ‘reported‘;
- Ihat – MAP incidence curve estimate;
- Chat – expected cases given MAP incidence curve estimate;
- LowCI – lower pointwise credible interval bands around the incidence curve; and
- HighCI – higher pointwise credible interval bands around the incidence curve.

Examples

```
indiana_model <- fit_incidence(
                  reported = spanish_flu$Indiana,
                  delay_dist = spanish_flu_delay_dist$proportion)
 indiana_df <- incidence_to_df(indiana_model, times = spanish_flu$Date)
```

---

init_params *Initialize spline parameters (beta)*

Description

Initialize spline parameters (beta) using a standard Gaussian distribution.

Usage

```
init_params(num_params)
```

Arguments

num_params	Integer size of desired parameter vector

Value

vector of size num_params

make_ar_extrap_samps *Make AR samples for extrapolation past end point*

Description

Make auto-regressive (AR) samples for extrapolation past end point to help with right-censoring problems.

Usage

```
make_ar_extrap_samps(
  reported,
  num_ar_steps = 10,
  num_ar_samps = 50,
  seed = 1,
  linear_tail = 14,
  extrapolation_prior_precision = 2
)
```

Arguments

reported	An integer vector of reported cases.
num_ar_steps	An integer number of AR steps after last observation.
num_ar_samps	An integer number of AR samples.
seed	Seed for RNG.
linear_tail	An integer number of days used to fit linear model on tail to be used as a mean for AR extrapolation.
extrapolation_prior_precision	A positive scalar for extrapolation slope shrinkage prior precision.

Value

A matrix of size (num_ar_samps x n + num_ar_steps)

make_likelihood_matrix *Make delay likelihood matrix*

Description

This function creates a matrix such that P[t, s] = P(C = t | I = s) = theta_t-s for s <= t and 0 otherwise.

Usage

```
make_likelihood_matrix(delay_dist)
```

Arguments

delay_dist	A positive vector that sums to one, which describes the delay distribution.

Value

A matrix of size n x n

---

make_spline_basis *Create spline basis matrix*

Description

This function creates basis matrix for spline model using cubic splines.

Usage

```
make_spline_basis(dof, tgrid)
```

Arguments

dof	An integer degrees of freedom.
tgrid	A grid of time values.

Value

A matrix of cubic spline basis values with ‘length(tgrid)‘ x ‘dof‘ entries.

---

marg_loglike_poisson *Marginal log likelihood This function computes the marginal probability of Pr(reported | beta). Note that lnPmat must be zero padded enough (or censored) to match the length of reported cases vector.*

Description

Marginal log likelihood This function computes the marginal probability of Pr(reported | beta). Note that lnPmat must be zero padded enough (or censored) to match the length of reported cases vector.

Usage

```
marg_loglike_poisson(beta, reported, Q, lnPmat)
```

Arguments

beta	spline parameter vector length num_params
reported	An integer vector of reported cases.
Q	spline basis matrix Tmod x num_params
lnPmat	matrix size Tobs x Tobs, log of make_likelihood_matrix

Value

A scalar log likelihood value.

---

marg_loglike_poisson_fisher

Marginal log likelihood Fisher information matrix

Description

This function computes the Fisher information matrix log likelihood term with respect to beta.

Usage

```
marg_loglike_poisson_fisher(beta, reported, Q, lnPmat)
```

Arguments

beta	A spline parameter vector length num_params.
reported	An integer vector of reported cases.
Q	A spline basis matrix Tmod x num_params.
lnPmat	A matrix size Tobs x Tobs, log of make_likelihood_matrix.

Value

A numeric vector, gradient of log likelihood value with respect to beta.

marg_loglike_poisson_grad

Marginal log likelihood gradient

Description

This function computes the gradient of the log likelihood term with respect to beta.

Usage

```
marg_loglike_poisson_grad(beta, reported, Q, lnPmat)
```

Arguments

beta	spline parameter vector length num_params
reported	An integer vector of reported cases.
Q	spline basis matrix Tmod x num_params
lnPmat	matrix size Tobs x Tobs, log of make_likelihood_matrix

Value

A numeric vector, gradient of log likelihood value with respect to beta.

plot.incidence_spline_model

Plot model from fit_incidence

Description

Plot time, reported cases, incidence curve with credible interval, and implied case curve.

Usage

```
## S3 method for class 'incidence_spline_model'
plot(x, ...)
```

Arguments

x	An "incidence_spline_model" output from fit_incidence.
...	Other parameters that can be included: • ‘times‘: an optional vector of time indices. • ‘plot_Chat‘: a logical for whether Chat should be plotted. • ‘plot_reported‘: a logical for whether reported cases should be plotted. • ‘plot_CI‘: a logical for whether CI should be plotted.

Examples

```
indiana_model <- fit_incidence(
                  reported = spanish_flu$Indiana,
                  delay_dist = spanish_flu_delay_dist$proportion)
 plot(indiana_model, times = spanish_flu$Date)
```

`poisson_objective` *Poisson objective function*

Description

This function computes Poisson objective function including regularizer.

Usage

```
poisson_objective(beta, lam, reported, Q, lnPmat, regularization_order)
```

Arguments

`beta`	spline parameter vector length num_params
`lam`	positive scalar regularization strength
`reported`	An integer vector of reported cases.
`Q`	spline basis matrix Tmod x num_params
`lnPmat`	matrix size Tobs x Tobs, log of make_likelihood_matrix
`regularization_order`	An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

Value

scalar objective function value

`poisson_objective_grad`
Poisson objective function gradient

Description

This function computes the Poisson objective function (including regularizer) gradient.

Usage

```
poisson_objective_grad(beta, lam, reported, Q, lnPmat, regularization_order)
```

Arguments

beta	spline parameter vector length num_params
lam	positive scalar regularization strength
reported	An integer vector of reported cases.
Q	spline basis matrix Tmod x num_params
lnPmat	matrix size Tobs x Tobs, log of make_likelihood_matrix
regularization_order	An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

Value

scalar objective function value

---

poisson_objective_post_cov_approx *Compute Fisher information matrix for Poisson objective*

Description

This function computes the Fisher information matrix for a regularized Poisson objective function.

Usage

```
poisson_objective_post_cov_approx(
  beta,
  lam,
  reported,
  Q,
  lnPmat,
  regularization_order
)
```

Arguments

beta	A vector of spline parameters.
lam	A regularization penalty parameter.
reported	A vector of reported values.
Q	A spline basis matrix.
lnPmat	A matrix size Tobs x Tobs, log of make_likelihood_matrix.
regularization_order	An integer that specifies the regularization order.

Value

Fisher information matrix of a regularized Poisson objective function.

regfun *Beta regularization function*

Description

This function computes regularization penalty term based on the betas and a difference.

Usage

```
regfun(beta, regularization_order = 2)
```

Arguments

beta	A spline parameter vector length num_params.
regularization_order	An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

Value

A scalar regularization value.

regfun_grad *Beta regularization function gradient*

Description

This function computes regularization penalty term gradient based on the betas and difference order.

Usage

```
regfun_grad(beta, regularization_order = 2)
```

Arguments

beta	spline parameter vector length num_params
regularization_order	An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

Value

scalar regularization value

regfun_hess *Beta regularization function Hessian*

Description

This function computes regularization penalty term Hessian based on the betas and differencing order.

Usage

```
regfun_hess(beta, regularization_order = 2)
```

Arguments

beta	spline parameter vector length num_params
regularization_order	An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

Value

scalar regularization value

sample_laplace_log_incidence_poisson *Generate Laplace samples of incidence*

Description

This function generates Laplace samples of posterior distribution for a vector of reported incidence.

Usage

```
sample_laplace_log_incidence_poisson(
  beta_hat,
  beta_cov,
  reported,
  Q,
  num_samps_per_ar = 10
)
```

Arguments

beta_hat	Maximum likelihood solution for beta parameter.
beta_cov	Covariance of objective solution (either Fisher information or Hessian inverse).
reported	An integer vector of reported cases.
Q	Spline basis matrix.
num_samps_per_ar	Number of Laplace samples to return for each AR path.

Value

A matrix of ‘num_samps_per_ar‘ log incidence curve samples from laplace approximation of distribution.

---

scan_spline_dof *Scan spline degrees of freedom*

Description

This function holds the regularization parameter value fixed and scans spline degrees of freedom.

Usage

```
scan_spline_dof(
  reported,
  delay_dist,
  dof_grid,
  method = "bic",
  lam = 0,
  regularization_order = 2,
  reported_val = NULL,
  end_pad_size = 0,
  fisher_approx_cov = FALSE
)
```

Arguments

reported	An integer vector of reported cases.
delay_dist	A positive vector that sums to one, which describes the delay distribution.
dof_grid	An integer vector of degrees of freedom for the spline basis.
method	Metric to choose "best" dof: ’aic’, ’bic’, ’val’. If method=’val’, reported_val must be non NULL and match reported size.
lam	A fixed value for the beta parameter regularization strength.
regularization_order	An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

`reported_val` Validation time series of equal size to reported vector for use with 'val' method. Default is NULL.

`end_pad_size` And integer number of steps the spline is defined beyond the final observation.

`fisher_approx_cov` A flag to use either the Fisher Information (TRUE) or the Hessian (FALSE) to approx posterior covariance over parameters.

Value

A list of degree of freedom fit statistics:

- best_dof = best degrees of freedom
- dof_resdf = data frame of fit statistics (lambda, dof, aic, bic, val_lls, train_lls)

`scan_spline_lam` *Scan spline regularization parameter*

Description

This function holds degrees of freedom fixed and scans regularization parameter values.

Usage

```
scan_spline_lam(
  reported,
  delay_dist,
  lam_grid,
  method = "val",
  percent_thresh = 2,
  dof = 10,
  regularization_order = 2,
  reported_val = NULL,
  end_pad_size = 0,
  fisher_approx_cov = TRUE
)
```

Arguments

`reported` An integer vector of reported cases.

`delay_dist` A positive vector that sums to one, which describes the delay distribution.

`lam_grid` A vector of regularization strengths to scan.

`method` Metric to choose "best" dof: 'aic', 'bic', 'val'. If method='val', reported_val must be non NULL and match reported size.

`percent_thresh` If using validation likelihood to select best, the largest (strongest) lambda that is within 'percent_thresh' of the highest validation lambda will be selected. Default is 2. Must be greater than 0.

`dof` Degrees of freedom for spline basis.

`regularization_order`
An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.

`reported_val` Validation time series of equal size to reported vector for use with 'val' method. Default is NULL.

`end_pad_size` And integer number of steps the spline is defined beyond the final observation.

`fisher_approx_cov`
A flag to use either the Fisher Information (TRUE) or the Hessian (FALSE) to approx posterior covariance over parameters.

Value

List of outputs:

- best_lam = best lambda
- lam_resdf = data frame of fit statistics (lambda, dof, aic, bic, val_lls, train_lls)

`spanish_flu` *Daily flu mortality from 1918 flu pandemic.*

Description

Daily mortality data from 1918-09-01 through 1918-12-31 in Indiana, Kansas, and Philadelphia

Usage

```
spanish_flu
```

Format

A data frame with 122 entries for 3 locations

Date date

Indiana daily deaths for all of Indiana

Kansas daily deaths for all of Kansas

Philadelphia daily deaths for Philadelphia

Source

Rogers SL (1920). Special Tables of Mortality from Influenza and Pneumonia, in Indiana, Kansas, and Philadelphia, PA (U.S. Dept Commerce, Washington, DC).

spanish_flu_delay_dist *Delay distribution from 1918 flu pandemic.*

Description

Daily death proportions.

Usage

```
spanish_flu_delay_dist
```

Format

A data frame with 31 entries and 3 columns.

days number of days since infection

proportion proportion of deaths that happen on that day

Source

Goldstein E, et al. (2009). Reconstructing influenza incidence by deconvolution of daily mortality time series (PNAS). https://www.pnas.org/content/pnas/106/51/21825.full.pdf

train_and_validate *Train and validate model on reported data*

Description

This function fit models with selected hyperparameters on reported data and return a matrix of posterior Laplace samples.

Usage

```
train_and_validate(
  reported,
  delay_dist,
  lam,
  dof,
  beta0 = NULL,
  regularization_order = 2,
  reported_val = NULL,
  end_pad_size = 0,
  fisher_approx_cov = TRUE,
  num_samps_per_ar = 10
)
```

Arguments

`reported`	An integer vector of reported cases.
`delay_dist`	A positive vector that sums to one, which describes the delay distribution.
`lam`	A fixed value for the beta parameter regularization strength.
`dof`	Degrees of freedom for spline basis.
`beta0`	(optional) Initial setting of spline parameters (before optimization)
`regularization_order`	An integer (typically 0, 1, 2), indicating differencing order for L2 regularization of spline parameters. Default is 2 for second derivative penalty.
`reported_val`	Validation time series of equal size to reported vector for use with 'val' method. Default is NULL.
`end_pad_size`	And integer number of steps the spline is defined beyond the final observation.
`fisher_approx_cov`	A flag to use either the Fisher Information (TRUE) or the Hessian (FALSE) to approx posterior covariance over parameters.
`num_samps_per_ar`	An integer for the number of Laplace samples per AR fit.

Value

A list of results of train and validate, including:

- train_ll = training log likelihood
- val_ll = validation log likelihood (if 'reported_val' is not 'NULL')
- Isamps = samples of the incidence curve from a Laplace approximation
- Ihat = MAP estimate of the incidence curve
- Chat = expected cases given MAP incidence curve
- beta_hat = MAP estimate of spline parameters
- beta_cov = covariance of spline parameters
- beta_hess = Hessian of spline parameters

`train_val_split` *Split reported case data*

Description

Split reported case integer time series into train and validate time series through thinning.

Usage

```
train_val_split(reported, frac_train = 0.75)
```

Arguments

`reported`	An integer vector of reported cases.
`frac_train`	A numeric between 0 and 1 for fraction of data used to train lambda validation.

Value

A list(reported_train, reported_val) where the elements reported_train and reported_val are both length, Tobs, and 'frac_train' of the counts fall in reported_train, the rest in reported_val.

Index