Package ‘fable.prophet’

August 20, 2020

Version 0.1.0

Title Prophet Modelling Interface for 'fable'

Description
Allows prophet models from the 'prophet' package to be used in a tidy workflow with the modelling interface of 'fabletools'. This extends 'prophet' to provide enhanced model specification and management, performance evaluation methods, and model combination tools.

Depends R (>= 3.1.3), Rcpp, fabletools (>= 0.2.0)

Imports rlang, tsibble, lubridate, prophet, dplyr, distributional

Suggests tsibbledata, testthat, ggplot2, covr, knitr, rmarkdown

ByteCompile true

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Language en-GB

URL https://pkg.mitchelloharawild.com/fable.prophet/

BugReports https://github.com/mitchelloharawild/fable.prophet/issues

VignetteBuilder knitr

NeedsCompilation no

Author Mitchell O'Hara-Wild [aut, cre],
Sean Taylor [ctb] (Prophet library,
https://facebook.github.io/prophet/),
Ben Letham [ctb] (Prophet library, https://facebook.github.io/prophet/)

Maintainer Mitchell O'Hara-Wild <mail@mitchelloharawild.com>

Repository CRAN

Date/Publication 2020-08-20 09:30:03 UTC

R topics documented:

fable.prophet-package 2
components.fbl_prophet 3
fitted.fbl_prophet 4
forecast.fbl_prophet 4
glance.fbl_prophet 5
prophet 6
residuals.fbl_prophet 8
tidy.fbl_prophet 8

Index **10**

`fable.prophet-package` *fable.prophet: Prophet Modelling Interface for 'fable'*

Description

Allows prophet models from the 'prophet' package to be used in a tidy workflow with the modelling interface of 'fabletools'. This extends 'prophet' to provide enhanced model specification and management, performance evaluation methods, and model combination tools.

Author(s)

Maintainer: Mitchell O'Hara-Wild <`mail@mitchelloharawild.com`>

Other contributors:

- Sean Taylor (Prophet library, https://facebook.github.io/prophet/) [contributor]
- Ben Letham (Prophet library, https://facebook.github.io/prophet/) [contributor]

See Also

Useful links:

- `https://pkg.mitchelloharawild.com/fable.prophet/`
- Report bugs at `https://github.com/mitchelloharawild/fable.prophet/issues`

`components.fbl_prophet`

Extract meaningful components

Description

A prophet model consists of terms which are additively or multiplicatively included in the model. Multiplicative terms are scaled proportionally to the estimated trend, while additive terms are not.

Usage

```
## S3 method for class 'fbl_prophet'
components(object, ...)
```

Arguments

`object`	An estimated model.
`...`	Unused.

Details

Extracting a prophet model's components using this function allows you to visualise the components in a similar way to `prophet::prophet_plot_components()`.

Value

A `fabletools::dable()` containing estimated states.

Examples

```
if (requireNamespace("tsibbledata")) {
library(tsibble)
beer_components <- tsibbledata::aus_production %>%
  model(
    prophet = prophet(Beer ~ season("year", 4, type = "multiplicative"))
  ) %>%
  components()

beer_components

autoplot(beer_components)

library(ggplot2)
library(lubridate)
beer_components %>%
  ggplot(aes(x = quarter(Quarter), y = year, group = year(Quarter))) +
  geom_line()
}
```

`fitted.fbl_prophet` *Extract fitted values*

Description

Extracts the fitted values from an estimated Prophet model.

Usage

```
## S3 method for class 'fbl_prophet'
fitted(object, ...)
```

Arguments

`object`	The time series model used to produce the forecasts
`...`	Additional arguments for forecast model methods.

Value

A vector of fitted values.

`forecast.fbl_prophet` *Produce forecasts from the prophet model*

Description

If additional future information is required (such as exogenous variables or carrying capacities) by the model, then they should be included as variables of the `new_data` argument.

Usage

```
## S3 method for class 'fbl_prophet'
forecast(object, new_data, specials = NULL, times = 1000, ...)
```

Arguments

`object`	The time series model used to produce the forecasts
`new_data`	A `tsibble` containing future information used to forecast.
`specials`	(passed by `fabletools::forecast.mdl_df()`).
`times`	The number of sample paths to use in estimating the forecast distribution when `boostrap = TRUE`.
`...`	Additional arguments passed to `prophet::predict.prophet()`.

Value

A list of forecasts.

See Also

`prophet::predict.prophet()`

Examples

```
if (requireNamespace("tsibbledata")) {
library(tsibble)
tsibbledata::aus_production %>%
  model(
    prophet = prophet(Beer ~ season("year", 4, type = "multiplicative"))
  ) %>%
  forecast()
}
```

`glance.fbl_prophet` *Glance a prophet model*

Description

A glance of a prophet provides the residual's standard deviation (sigma), and a tibble containing the selected changepoints with their trend adjustments.

Usage

```
## S3 method for class 'fbl_prophet'
glance(x, ...)
```

Arguments

`x`	model or other R object to convert to single-row data frame
`...`	other arguments passed to methods

Value

A one row tibble summarising the model's fit.

Examples

```
if (requireNamespace("tsibbledata")) {
library(tsibble)
library(dplyr)
fit <- tsibbledata::aus_production %>%
  model(
    prophet = prophet(Beer ~ season("year", 4, type = "multiplicative"))
  )

glance(fit)
}
```

`prophet` *Prophet procedure modelling*

Description

Prepares a prophet model specification for use within the `fable` package.

Usage

```
prophet(formula, ...)
```

Arguments

`formula`	A symbolic description of the model to be fitted of class `formula`.
`...`	Additional arguments passed to the `optimizing` or `sampling` functions in Stan.

Details

The prophet modelling interface uses a `formula` based model specification (`y ~ x`), where the left of the formula specifies the response variable, and the right specifies the model's predictive terms. Like any model in the fable framework, it is possible to specify transformations on the response.

A prophet model supports piecewise linear or exponential growth (trend), additive or multiplicative seasonality, holiday effects and exogenous regressors. These can be specified using the 'specials' functions detailed below. The introduction vignette provides more details on how to model data using this interface to prophet: `vignette("intro",package="fable.prophet")`.

Specials

growth: The `growth` special is used to specify the trend parameters.

```
growth(type = c("linear", "logistic"), capacity = NULL, floor = NULL,
       changepoints = NULL, n_changepoints = 25, changepoint_range = 0.8,
       changepoint_prior_scale = 0.05)
```

`type`	The type of trend (linear or logistic).
`capacity`	The carrying capacity for when `type` is "logistic".
`floor`	The saturating minimum for when `type` is "logistic".
`changepoints`	A vector of dates/times for changepoints. If `NULL`, changepoints are automatically selected.
`n_changepoints`	The total number of changepoints to be selected if `changepoints` is `NULL`
`changepoint_range`	Proportion of the start of the time series where changepoints are automatically selected.
`changepoint_prior_scale`	Controls the flexibility of the trend.

season: The `season` special is used to specify a seasonal component. This special can be used multiple times for different seasonalities.

Warning: The inputs controlling the seasonal `period` is specified is different than `prophet::prophet()`. Numeric inputs are treated as the number of observations in each seasonal period, not the number of days.

```
season(period = NULL, order = NULL, prior_scale = 10,
       type = c("additive", "multiplicative"), name = NULL)
```

`period`	The periodic nature of the seasonality. If a number is given, it will specify the number of observations in each
`order`	The number of terms in the partial Fourier sum. The higher the `order`, the more flexible the seasonality can b
`prior_scale`	Used to control the amount of regularisation applied. Reducing this will dampen the seasonal effect.
`type`	The nature of the seasonality. If "additive", the variability in the seasonal pattern is fixed. If "multiplicative", t
`name`	The name of the seasonal term (allowing you to name an annual pattern as 'annual' instead of 'year' or `365.2`

holiday: The `holiday` special is used to specify a `tsibble` containing holidays for the model.

```
holiday(holidays = NULL, prior_scale = 10L)
```

`holidays`	A `tsibble` containing a set of holiday events. The event name is given in the 'holiday' column, and the event
`prior_scale`	Used to control the amount of regularisation applied. Reducing this will dampen the holiday effect.

xreg: The `xreg` special is used to include exogenous regressors in the model. This special can be used multiple times for different regressors with different arguments. Exogenous regressors can also be used in the formula without explicitly using the `xreg()` special, which will then use the default arguments.

```
xreg(..., prior_scale = NULL, standardize = "auto", type = NULL)
```

`...`	A set of bare expressions that are evaluated as exogenous regressors
`prior_scale`	Used to control the amount of regularisation applied. Reducing this will dampen the regressor effect.
`standardize`	Should the regressor be standardised before fitting? If "auto", it will standardise if the regressor is not binary.
`type`	Does the effect of the regressor vary proportionally to the level of the series? If so, "multiplicative" is best. Ot

See Also

- `prophet::prophet()`
- Prophet homepage

- Prophet R package
- Prophet Python package

Examples

```
library(tsibble)
as_tsibble(USAccDeaths) %>%
  model(
    prophet = prophet(value ~ season("year", 4, type = "multiplicative"))
  )
```

residuals.fbl_prophet *Extract model residuals*

Description

Extracts the residuals from an estimated Prophet model.

Usage

```
## S3 method for class 'fbl_prophet'
residuals(object, ...)
```

Arguments

`object`	The time series model used to produce the forecasts
`...`	Additional arguments for forecast model methods.

Value

A vector of residuals.

tidy.fbl_prophet *Extract estimated coefficients from a prophet model*

Description

Extract estimated coefficients from a prophet model

Usage

```
## S3 method for class 'fbl_prophet'
tidy(x, ...)
```

Arguments

`x`	An object to be converted into a tidy `tibble::tibble()`.
`...`	Additional arguments to tidying method.

Value

A tibble containing the model's estimated parameters.

Examples

```
if (requireNamespace("tsibbledata")) {
library(tsibble)
library(dplyr)
fit <- tsibbledata::aus_production %>%
  model(
    prophet = prophet(Beer ~ season("year", 4, type = "multiplicative"))
  )

tidy(fit) # coef(fit) or coefficients(fit) can also be used
}
```

Index

∗ **package**
    fable.prophet-package, 2

components.fbl_prophet, 3

fable.prophet *(*fable.prophet-package*)*, 2
fable.prophet-package, 2
fabletools::dable(), *3*
fabletools::forecast.mdl_df(), *4*
fitted.fbl_prophet, 4
forecast.fbl_prophet, 4

glance.fbl_prophet, 5

prophet, 6
prophet::predict.prophet(), *4, 5*
prophet::prophet(), *7*
prophet::prophet_plot_components(), *3*

residuals.fbl_prophet, 8

tibble::tibble(), *9*
tidy.fbl_prophet, 8