Package 'unfold'

April 15, 2026

Type Package

Title Mapping Hidden Geometry into Future Sequences

Version 1.0.1

Maintainer Giancarlo Vercellino <giancarlo.vercellino@gmail.com>

Description A variational mapping approach that reveals and expands future temporal dynamics from folded high-dimensional geometric distance spaces, unfold turns a set of time series into a 4D block of pairwise distances between reframed windows, learns a variational mapper that maps those distances to the next reframed window, and produces horizon-wise predictive functions for each input series. In short: it unfolds the future path of each series from a folded geometric distance representation.

License GPL-3

LazyData true

RoxygenNote 7.3.3

Imports torch (>= 0.11.0), purrr (>= 1.0.1), imputeTS (>= 3.3), lubridate (>= 1.9.2), ggplot2 (>= 3.5.1), scales (>= 1.3.0), abind (>= 1.4-5), coro (>= 1.1.0)

Encoding UTF-8

URL https://rpubs.com/giancarlo_vercellino/unfold

Suggests knitr, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1.0)

NeedsCompilation no

Author Giancarlo Vercellino [aut, cre, cph]

Repository CRAN

Date/Publication 2026-04-15 05:10:02 UTC

Contents

dummy_set *Tech Stock Time Series Dataset*

Description

A multivariate dataset for closing prices for several major tech stocks over time. Source: YahooFinance.

Usage

```
data(dummy_set)
```

Format

A data frame with 2133 observations of 4 variables:

dates Character vector of dates in "YYYY-MM-DD" format.

TSLA.Close Numeric. Closing prices for Tesla.

MSFT.Close Numeric. Closing prices for Microsoft.

MARA.Close Numeric. Closing prices for MARA Holdings.

Examples

```
data(dummy_set)
plot(as.Date(dummy_set$dates), dummy_set$TSLA.Close, type = "l")
```

unfold *unfold: Mapping Hidden Geometry into Future Sequences*

Description

A variational mapping approach that reveals and expands future temporal dynamics from folded high-dimensional geometric distance spaces, unfold turns a set of time series into a 4D block of pairwise distances between reframed windows, learns a variational mapper that maps those distances to the next reframed window, and produces horizon-wise predictive functions for each input series. In short: it unfolds the future path of each series from a folded geometric distance representation.

Usage

```
unfold(
  ts_set,
  horizon,
  metric = "euclidean",
  latent_dim = 32,
  enc_hidden = c(512, 256),
  dec_hidden = c(256, 512),
  p_drop = 0.1,
  out_kind = "linear",
  epochs = 30,
  batch_size = 64,
  lr = 0.001,
  beta = 1,
  beta_warmup = 0,
  grad_clip = NULL,
  valid_split = 0.2,
  verbose = TRUE,
  alpha = 0.1,
  dates = NULL,
  patience = NULL,
  n_bases = 10,
  seed = 42
)
```

Arguments

`ts_set`	A data frame containing the time series, one column per series.
`horizon`	Integer. Forecast horizon; controls reframing and output functions.
`metric`	Distance metric fro the 4D tensor; one of "euclidean", "mahalanobis", "cosine". Default: "euclidean".
`latent_dim`	Integer. Latent dimensionality of the variational mapper. Default: 32.
`enc_hidden, dec_hidden`	Integer vectors. Hidden layer widths for encoder/decoder MLPs, defaulting to c(512, 256) and c(256, 512) respectively.
`p_drop`	Dropout probability in encoder/decoder. Default: 0.1.
`out_kind`	Output nonlinearity of the decoder; one of "linear", "tanh" (used by the VAM). Default: "linear".
`epochs`	Integer. Training epochs. Default: 30.
`batch_size`	Integer. Dimension of batch. Default: 64.
`lr`	Double. Learning rate. Default: 1e-3.
`beta`	Double. KL weight for the variational objective. Default: 1.
`beta_warmup`	Integer. If > 0, linearly warm up beta over this many epochs. Default: 0.
`grad_clip`	Optional max norm for gradient clipping. If you never see exploding losses or NaNs, you can leave it NULL, otherwise, if training diverges, try clipping (1 to 5) and monitor if loss becomes smoother. Default: NULL.

valid_split	Double. Proportion of samples held out for validation during VAM training. Default: 0.2.
verbose	Logical. Print training progress. Default: TRUE.
alpha	Double. Forecasting confidence interval used in plotted graphs. Default: 0.1.
dates	Character. Vector with the original time series dates in text format, used for plotting purposes. Default: NULL.
patience	Integer Epochs of stagnation before early stopping. Default: NULL.
n_bases	Integer Maximum number of distributions to use for the Gaussian mixture. Default: 10.
seed	Random seed for reproducibility. Default: 42.

Value

A named list with the following components:

'description' Character string giving a short description of the model (parameters, activations and so on).

'model' A fitted variational mapper object of class vam_fit. This object contains the trained network plus helper methods (encode, decode, reconstruct, predict, etc.).

'dist_array' A numeric 4D array containing pairwise distances between reframed time-series windows: shape N x N x M x M, where N is the number of reframed time-series windows and M the number of time series.

'loss_plot' A ggplot plot object showing the training and validation loss curves across epochs.

'pred_funs' For each time series, a length-horizon list containing four gaussian mix distribution functions (dfun, pfun, qfun, rfun).

'graph_plot' A list including ggplot graphs for each time series reproducing the predicted horizon with confidence interval alpha.

'time_log' An object measuring the elapsed time for the computation (preprocessing, training, prediction, etc.).

Author(s)

Maintainer: Giancarlo Vercellino <giancarlo.vercellino@gmail.com> [copyright holder]

See Also

Useful links:

- https://rpubs.com/giancarlo_vercellino/unfold

Examples

```
.has_working_torch <- function() {
  if (!requireNamespace("torch", quietly = TRUE)) return(FALSE)
  tryCatch({
    torch::torch_tensor(1)$item()
    TRUE
```

```
  }, error = function(e) FALSE)
}

if (.has_working_torch()) {
  set.seed(42)
  TT <- 100
  ts_set <- data.frame(
    A = cumsum(rnorm(TT, 0.02, 0.10)) + 10,
    B = cumsum(rnorm(TT, 0.01, 0.08)) +  8,
    C = cumsum(rnorm(TT, 0.00, 0.12)) + 12
  )

  fit <- unfold(
    ts_set = ts_set,
    horizon = 3,
    epochs = 5,
    batch_size = 16,
    verbose = FALSE
  )

  names(fit$pred_funs)
}
```

Index