Package: ECTSVR (via r-universe)

November 13, 2026

Title Error Correction Term Based Support Vector Regression for Time Series

Version 0.2.0

Maintainer Pankaj Das <pankaj.iasri@gmail.com>

Description The cointegration based support vector regression model is a combination of error correction model and support vector regression. This hybrid model allows the researcher to make use of the information extracted by the cointegrating vector as an input in the support vector regression mode. Implements the models with multiple kernel functions (linear, polynomial, radial, sigmoid) for time series forecasting. Includes stationarity tests, Johansen cointegration, and comprehensive model evaluation.

Depends R (>= 3.5.0)

Imports urca, tseries, e1071, caret, Metrics, knitr, stats, graphics

Suggests rmarkdown, testthat

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Pankaj Das [aut, cre] (ORCID: <https://orcid.org/0000-0003-1672-2502>)

Config/pak/sysreqs libicu-dev libssl-dev

Repository https://pankajdas7.r-universe.dev

Date/Publication 2026-11-12 16:40:15 UTC

RemoteUrl https://github.com/cran/ECTSVR

RemoteRef HEAD

RemoteSha a4eefec6f09a89125069094a8f2a954ffa9374fb

Contents

calculate_metrics 2
evaluate_models 3
plot_comparisons 4
prepare_ectsvr_data 5
run_ectsvr_analysis 6
test_cointegration 7
test_stationarity 8
train_svr_models 9

Index **11**

`calculate_metrics` *Calculate Performance Metrics*

Description

This function calculates common performance metrics for regression models including RMSE, MAE, MAPE, and R-squared.

Usage

```
calculate_metrics(actual, predicted)
```

Arguments

`actual`	A numeric vector of actual values
`predicted`	A numeric vector of predicted values

Value

A named numeric vector containing:

`RMSE`	Root Mean Square Error
`MAE`	Mean Absolute Error
`MAPE`	Mean Absolute Percentage Error
`R2`	R-squared (coefficient of determination)

Examples

```
actual <- c(10, 12, 15, 18, 20)
predicted <- c(11, 13, 14, 17, 21)
metrics <- calculate_metrics(actual, predicted)
print(metrics)
```

`evaluate_models` *Compare and Evaluate Models*

Description

This function compares standard SVR and ECT-SVR models, identifies the best performing models based on different metrics, and creates a comparison table.

Usage

```
evaluate_models(std_results, ect_results, format = "simple", verbose = TRUE)
```

Arguments

`std_results`	A list containing results from standard SVR models
`ect_results`	A list containing results from ECT-SVR models
`format`	Output format for the table when using knitr. Options: "simple", "markdown", "latex", "html", or "pipe" (default: "simple")
`verbose`	Logical; print the summary and table (default: TRUE)

Value

A list containing:

`combined_results`
: Data frame with combined results from both model types

`best_models`
: List of best models by RMSE, MAE, MAPE, and R2

`comparison_table`
: Formatted kable table for display (if knitr available)

Examples

```
## Not run:
# Assuming you have trained models
comparison <- evaluate_models(std_results, ect_results)
print(comparison$combined_results)
print(comparison$best_models$best_rmse)

# With markdown format
comparison <- evaluate_models(std_results, ect_results, format = "markdown")

## End(Not run)
```

`plot_comparisons` *Plot Model Comparisons*

Description

This function creates visual comparisons between actual values, standard SVR predictions, and ECT-SVR predictions for different kernel types.

Usage

```
plot_comparisons(
  y_test,
  std_predictions,
  ect_predictions,
  kernels = c("linear", "polynomial", "radial", "sigmoid"),
  ylab = "Fruit Price"
)
```

Arguments

`y_test`	A numeric vector of actual test values
`std_predictions`	A list containing standard SVR predictions for each kernel
`ect_predictions`	A list containing ECT-SVR predictions for each kernel
`kernels`	A character vector of kernel types (default: c("linear", "polynomial", "radial", "sigmoid"))
`ylab`	Label of the y axis (default: "Fruit Price")

Value

No return value, creates plots in the current graphics device

Examples

```
## Not run:
# After training models
plot_comparisons(
  data$y_test,
  std_results$predictions,
  ect_results$predictions
)

## End(Not run)
```

`prepare_ectsvr_data` *Prepare Data for ECT-SVR Analysis*

Description

This function prepares time series data for ECT-SVR analysis by performing Johansen cointegration test, constructing error correction terms, creating lag features, and splitting data into training and test sets.

Usage

```
prepare_ectsvr_data(
  fruit,
  oil,
  train_ratio = 0.8,
  lags = 2,
  ect_sample = c("train", "full")
)
```

Arguments

`fruit`	A numeric vector of fruit prices
`oil`	A numeric vector of crude oil prices
`train_ratio`	A numeric value between 0 and 1 specifying the proportion of data to use for training (default: 0.8)
`lags`	Number of lags of `fruit` used as features (default: 2). Must be at least 1.
`ect_sample`	Sample used to estimate the cointegrating vector: `"train"` (default; avoids look-ahead into the test period) or `"full"`.

Value

A list containing:

`train`	Training dataset as data frame
`test`	Test dataset as data frame
`X_train_std`	Standard features for training (lags only)
`X_test_std`	Standard features for testing (lags only)
`X_train_ect`	ECT-enhanced features for training
`X_test_ect`	ECT-enhanced features for testing
`y_train`	Training target values
`y_test`	Test target values
`ECT`	Error correction term vector
`johansen_test`	Johansen test result object
`beta`	Cointegrating vector

Examples

```
## Not run:
# Generate example data
fruit <- rnorm(100, 100, 10)
oil <- rnorm(100, 50, 5)

# Prepare data
data <- prepare_ectsvr_data(fruit, oil)
str(data)

## End(Not run)
```

`run_ectsvr_analysis` *Main Function for ECT-SVR Analysis*

Description

This function performs a complete ECT-SVR analysis including stationarity tests, cointegration tests, model training with multiple kernels, and comprehensive evaluation. It serves as the main entry point for the package.

Usage

```
run_ectsvr_analysis(
  fruit,
  oil,
  train_ratio = 0.8,
  kernels = c("linear", "polynomial", "radial", "sigmoid"),
  verbose = TRUE,
  plot_results = TRUE,
  lags = 2
)
```

Arguments

`fruit`	A numeric vector of fruit prices
`oil`	A numeric vector of crude oil prices
`train_ratio`	A numeric value between 0 and 1 for training proportion (default: 0.8)
`kernels`	A character vector of kernel types to use. Options are "linear", "polynomial", "radial", and "sigmoid" (default: all four)
`verbose`	A logical value indicating whether to print detailed output (default: TRUE)
`plot_results`	A logical value indicating whether to generate comparison plots (default: TRUE)
`lags`	Number of lags of the fruit series used as features (default: 2)

Value

A list containing all analysis results:

stationarity	Results from stationarity tests
cointegration	Results from cointegration tests
data	Prepared data for modeling
std_svr	Standard SVR model results
ect_svr	ECT-SVR model results
evaluation	Model evaluation and comparison results
best_model	Information about the best performing model

Examples

```
## Not run:
# Load your data
fruit <- c(100, 102, 101, 105, 108, 107, 110, 112, 111, 115)
oil <- c(50, 51, 50.5, 52, 51.5, 53, 52.5, 54, 53.5, 55)

# Run complete analysis
results <- run_ectsvr_analysis(fruit, oil)

# View best model
print(results$best_model)

# View comparison table
print(results$evaluation$comparison_table)

## End(Not run)
```

---

test_cointegration *Perform Johansen Cointegration Test*

Description

This function performs the Johansen cointegration test on fruit and oil price series and extracts the cointegrating vector and error correction term.

Usage

```
test_cointegration(fruit, oil, verbose = TRUE)
```

Arguments

fruit	A numeric vector of fruit prices
oil	A numeric vector of crude oil prices
verbose	Logical; print the test summary and cointegrating vector (default: TRUE)

Value

A list containing:

johansen_test	The Johansen test result object from urca::ca.jo
beta	The first cointegrating vector
ECT	The error correction term calculated from the cointegrating vector

Examples

```
## Not run:
fruit <- rnorm(100, 100, 10)
oil <- rnorm(100, 50, 5)
coint <- test_cointegration(fruit, oil, verbose = TRUE)
print(coint$beta)
plot(coint$ECT, type = "l")

## End(Not run)
```

---

test_stationarity *Perform Stationarity Tests on Time Series*

Description

This function performs various stationarity tests including Jarque-Bera, Augmented Dickey-Fuller (ADF), and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests on oil and fruit price series.

Usage

```
test_stationarity(oil, fruit, verbose = TRUE)
```

Arguments

oil	A numeric vector of oil prices
fruit	A numeric vector of fruit prices
verbose	Logical; print the test results (default: TRUE)

Value

A list containing test results:

jb_oil	Jarque-Bera test result for oil (from tseries::jarque.bera.test)
jb_fruit	Jarque-Bera test result for fruit
adf_oil	ADF test result for oil (from tseries::adf.test)
kpss_oil	KPSS test result for oil (from tseries::kpss.test)
adf_fruit	ADF test result for fruit
kpss_fruit	KPSS test result for fruit

Examples

```
## Not run:
fruit <- rnorm(100, 100, 10)
oil <- rnorm(100, 50, 5)
results <- test_stationarity(oil, fruit, verbose = TRUE)
print(results$adf_oil)

## End(Not run)
```

`train_svr_models` *Train SVR Models with Different Kernels*

Description

This function trains Support Vector Regression models using different kernel functions, with options for hyperparameter tuning. It scales the features, trains models for each specified kernel, and returns performance metrics.

Usage

```
train_svr_models(
  X_train,
  X_test,
  y_train,
  y_test,
  model_name,
  kernels = c("linear", "polynomial", "radial", "sigmoid"),
  cost = 100,
  gamma = 0.1,
  epsilon = 0.1,
  verbose = TRUE
)
```

Arguments

`X_train`	A data frame or matrix of training features
`X_test`	A data frame or matrix of test features
`y_train`	A numeric vector of training target values
`y_test`	A numeric vector of test target values
`model_name`	A character string naming the model type (e.g., "Standard SVR")
`kernels`	A character vector of kernel types. Options: "linear", "polynomial", "radial", "sigmoid" (default: all four)
`cost`	A numeric value for the cost parameter C (default: 100)
`gamma`	A numeric value for the gamma parameter (default: 0.1)
`epsilon`	A numeric value for the epsilon parameter (default: 0.1)
`verbose`	Logical; print progress messages (default: TRUE)

Value

A list containing:

`results`	A data frame with performance metrics for each kernel
`predictions`	A list of predictions for training and test sets for each kernel
`models`	A list of trained SVR model objects for each kernel

Examples

```
## Not run:
# Assuming prepared data from prepare_ectsvr_data()
data <- prepare_ectsvr_data(fruit, oil)
results <- train_svr_models(
  data$X_train_std,
  data$X_test_std,
  data$y_train,
  data$y_test,
  "Standard SVR"
)
print(results$results)

## End(Not run)
```

Index

calculate_metrics, 2

evaluate_models, 3

plot_comparisons, 4
prepare_ectsvr_data, 5

run_ectsvr_analysis, 6

test_cointegration, 7
test_stationarity, 8
train_svr_models, 9