Package 'SCE'

May 11, 2026

Title Stepwise Clustered Ensemble

Version 1.1.4

Description Implementation of Stepwise Clustered Ensemble (SCE) and Stepwise Cluster Analysis (SCA) for multivariate data analysis. The package provides comprehensive tools for feature selection, model training, prediction, and evaluation in hydrological and environmental modeling applications. Key functionalities include recursive feature elimination (RFE), Wilks feature importance analysis, model validation through out-of-bag (OOB) validation, and ensemble prediction capabilities. The package supports both single and multivariate response variables, making it suitable for complex environmental modeling scenarios. For more details see Li et al. (2021) <doi:10.5194/hess-25-4947-2021>.

URL https://doi.org/10.5194/hess-25-4947-2021

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 3.5.0)

Imports stats (>= 3.5.0), utils (>= 3.5.0)

Suggests testthat (>= 3.0.0), knitr, rmarkdown

NeedsCompilation no

Author Kailong Li [aut, cre]

Maintainer Kailong Li <lkl98509509@gmail.com>

Repository CRAN

Date/Publication 2026-05-11 07:50:06 UTC

Contents

air_quality_datasets 2
evaluate 3
importance 4
plot_rfe 4
predict 5
print 6

rfe_sce 7
sca 8
sce 9
streamflow_datasets 10
summary 12

Index **13**

`air_quality_datasets` *Air Quality Dataset*

Description

These datasets contain air quality measurements for training and testing purposes. They include various air pollutant concentrations and meteorological variables measured at different locations and times.

Usage

```
data("air_quality_training")
data("air_quality_testing")
```

Format

Both datasets are data frames with 8760 rows and 12 variables:

Date Date and time of measurement (POSIXct format)

PM2.5 Particulate matter with diameter less than 2.5 micrometers (μg/m^3)

PM10 Particulate matter with diameter less than 10 micrometers (μg/m^3)

SO2 Sulfur dioxide concentration (μg/m^3)

NO2 Nitrogen dioxide concentration (μg/m^3)

CO Carbon monoxide concentration (μg/m^3)

O3 Ozone concentration (μg/m^3)

TEMP Temperature (°C)

PRES Atmospheric pressure (hPa)

DEWP Dew point temperature (°C)

RAIN Precipitation amount (mm)

WSPM Wind speed (m/s)

Details

Dataset Differences:

- `air_quality_training`: Used for training SCA and SCE models
- `air_quality_testing`: Used for testing trained models

Variable Descriptions:

- **PM2.5, PM10**: Particulate matter concentrations, important indicators of air quality
- **SO2, NO2, CO, O3**: Major air pollutants regulated by environmental agencies
- **TEMP, PRES, DEWP**: Meteorological variables affecting air quality
- **RAIN, WSPM**: Weather conditions that influence pollutant dispersion

Source

Air quality monitoring stations

`evaluate` *Evaluate SCE and SCA Model Performance*

Description

Evaluate model performance for SCE or SCA models.

Usage

```
## S3 method for class 'sce'
evaluate(object, testing_data, training_data, digits = 3, ...)
## S3 method for class 'sca'
evaluate(object, testing_data, training_data, digits = 3, ...)
```

Arguments

`object`	An SCE or SCA model object
`testing_data`	Testing dataset
`training_data`	Training dataset
`digits`	Number of decimal places (default: 3)
`...`	Additional arguments

Value

Model performance metrics.

See Also

`sce`, `sca`, `predict`

importance *Variable Importance for SCE and SCA Models*

Description

Calculate variable importance for SCE or SCA models.

Usage

```
## S3 method for class 'sce'
importance(object, oob_weight = TRUE, digits = 2, ...)
## S3 method for class 'sca'
importance(object, digits = 2, ...)
```

Arguments

`object`	An SCE or SCA model object
`oob_weight`	Use out-of-bag weights for importance calculation (SCE only, default: TRUE)
`digits`	Number of decimal places to round the returned relative importance values (default: 2)
`...`	Additional arguments

Value

Variable importance rankings. For convenience, relative importance values are rounded to `digits` decimal places.

See Also

`sce`, `sca`, `rfe_sce`

plot_rfe *Plot Recursive Feature Elimination Results*

Description

Plot Recursive Feature Elimination results.

Usage

```
plot_rfe(rfe_result,
         main = "OOB Validation and Testing R2 vs Number of Predictors",
         col_validation = "blue",
         col_testing = "red",
         pch = 16,
         lwd = 2,
         cex = 1.2,
         legend_pos = "bottomleft",
         ...)
```

Arguments

`rfe_result`	Result object from `rfe_sce` function
`main`	Plot title
`col_validation`	Color for validation line
`col_testing`	Color for testing line
`pch`	Point character
`lwd`	Line width
`cex`	Point size
`legend_pos`	Legend position
`...`	Additional arguments

Value

Plot showing validation and testing R2 vs number of predictors.

See Also

`rfe_sce`

`predict` *Predict Using SCE and SCA Models*

Description

Make predictions on new data using SCE or SCA models.

Usage

```
## S3 method for class 'sce'
predict(object, newdata, ...)
## S3 method for class 'sca'
predict(object, newdata, ...)
```

Arguments

object	An SCE or SCA model object
newdata	New data for prediction
...	Additional arguments

Value

Predictions for the new data.

See Also

sce, sca, evaluate

---

print *Print SCE and SCA Model Objects*

Description

Print information about SCE or SCA model objects.

Usage

```
## S3 method for class 'sce'
print(x, ...)
## S3 method for class 'sca'
print(x, ...)
```

Arguments

x	An SCE or SCA model object
...	Additional arguments (not used)

Details

For SCE objects, prints ensemble information including number of trees, parameters, predictors, predictants, and OOB performance metrics.

For SCA objects, prints tree structure information including total nodes, leaf nodes, cutting/merging actions, and variable names.

Value

Prints model information and returns the object invisibly.

See Also

sce, sca, summary

rfe_sce *Recursive Feature Elimination for SCE Models*

Description

Recursive Feature Elimination for SCE models to identify the most important predictors.

Usage

```
rfe_sce(training_data, testing_data, predictors, predictant, nmin, ntree,
        alpha = 0.05, resolution = 1000, step = 1, verbose = TRUE,
        parallel = TRUE)
```

Arguments

training_data	Training dataset
testing_data	Testing dataset
predictors	Character vector of predictor names
predictant	Character vector of predictant names
nmin	Minimum samples per node
ntree	Number of trees
alpha	Significance level (default: 0.05)
resolution	Resolution for splitting (default: 1000)
step	Number of predictors to remove per iteration (default: 1)
verbose	Print progress (default: TRUE)
parallel	Use parallel processing (default: TRUE)

Value

RFE results with performance metrics and importance scores.

See Also

plot_rfe, sce, importance

sca *Stepwise Cluster Analysis (SCA)*

Description

Builds a single Stepwise Cluster Analysis (SCA) tree model that recursively partitions the data space based on Wilks' Lambda statistic.

Usage

```
sca(training_data, x, y, nmin, alpha = 0.05, resolution = 1000, verbose = FALSE)
```

Arguments

training_data	A data.frame containing the training data
x	Character vector of predictor variable names
y	Character vector of predictant variable names
nmin	Minimum number of samples in a leaf node
alpha	Significance level for clustering (default: 0.05)
resolution	Resolution for splitting (default: 1000)
verbose	Print progress information (default: FALSE)

Value

An S3 object of class "sca" containing the tree model.

See Also

sce, predict, importance, evaluate

Examples

```
# Load example data
data(streamflow_training_10var)
data(streamflow_testing_10var)

# Define variables
Predictors <- c("Prcp","SRad","Tmax","Tmin","VP","smlt","swvl1","swvl2","swvl3","swvl4")
Predictants <- c("Flow")

# Build SCA model
sca_model <- sca(
  training_data = streamflow_training_10var,
  x = Predictors,
  y = Predictants,
  nmin = 5,
  alpha = 0.05,
```

```
  resolution = 1000
)

# Use S3 methods
print(sca_model)
summary(sca_model)
sca_predictions <- predict(sca_model, streamflow_testing_10var)
sca_importance <- importance(sca_model)
sca_evaluation <- evaluate(sca_model, streamflow_testing_10var, streamflow_training_10var)
```

`sce` *Stepwise Clustered Ensemble (SCE)*

Description

Builds a Stepwise Clustered Ensemble (SCE) model, which is an ensemble of SCA trees built using bootstrap samples and random feature selection, providing improved prediction accuracy and robustness.

Usage

```
sce(training_data, x, y, mfeature, nmin, ntree, alpha = 0.05,
    resolution = 1000, verbose = FALSE, parallel = TRUE)
```

Arguments

`training_data`	A data.frame containing the training data
`x`	Character vector of predictor variable names
`y`	Character vector of predictant variable names
`mfeature`	Number of features to randomly select for each tree
`nmin`	Minimum number of samples in a leaf node
`ntree`	Number of trees in the ensemble
`alpha`	Significance level for clustering (default: 0.05)
`resolution`	Resolution for splitting (default: 1000)
`verbose`	Print progress information (default: FALSE)
`parallel`	Use parallel processing (default: TRUE)

Value

An S3 object of class `"sce"` containing the ensemble model.

See Also

`sca`, `predict`, `importance`, `evaluate`

Examples

```
# Load example data
data(streamflow_training_10var)
data(streamflow_testing_10var)

# Define variables
Predictors <- c("Prcp","SRad","Tmax","Tmin","VP","smlt","swvl1","swvl2","swvl3","swvl4")
Predictants <- c("Flow")

# Build SCE model
sce_model <- sce(
  training_data = streamflow_training_10var,
  x = Predictors,
  y = Predictants,
  mfeature = round(0.5 * length(Predictors)),
  nmin = 5,
  ntree = 48,
  alpha = 0.05,
  resolution = 1000,
  parallel = FALSE
)

# Use S3 methods
print(sce_model)
summary(sce_model)
sce_predictions <- predict(sce_model, streamflow_testing_10var)
sce_importance <- importance(sce_model)
sce_evaluation <- evaluate(sce_model, streamflow_testing_10var, streamflow_training_10var)
```

`streamflow_datasets` *Streamflow Dataset*

Description

These datasets contain streamflow and related environmental variables for training and testing purposes. They are used in examples to demonstrate the SCE package functionality with different levels of complexity.

Usage

```
data("streamflow_training_10var")
data("streamflow_training_22var")
data("streamflow_testing_10var")
data("streamflow_testing_22var")
```

Format

streamflow_training_10var: Basic environmental variables (12 columns):

Date Date and time of measurement
Prcp Monthly mean daily precipitation (mm)
SRad Monthly mean daily solar radiation (W/m^2)
Tmax Monthly mean daily maximum temperature (°C)
Tmin Monthly mean daily minimum temperature (°C)
VP Monthly mean daily vapor pressure (Pa)
smlt Monthly snowmelt (m)
swvl1 Soil water content layer 1 (m^3/m^3)
swvl2 Soil water content layer 2 (m^3/m^3)
swvl3 Soil water content layer 3 (m^3/m^3)
swvl4 Soil water content layer 4 (m^3/m^3)
Flow Monthly mean daily streamflow (cfs)

streamflow_training_22var: Extended variables with climate indices (24 columns):

Flow Streamflow measurements
IPO Interdecadal Pacific Oscillation
IPO_lag1 IPO with 1-month lag
IPO_lag2 IPO with 2-month lag
Nino3.4 Nino 3.4 index
Nino3.4_lag1 Nino 3.4 with 1-month lag
Nino3.4_lag2 Nino 3.4 with 2-month lag
PDO Pacific Decadal Oscillation
PDO_lag1 PDO with 1-month lag
PDO_lag2 PDO with 2-month lag
PNA Pacific North American pattern
PNA_lag1 PNA with 1-month lag
PNA_lag2 PNA with 2-month lag
Precipitation Monthly precipitation
Precipitation_2Mon 2-month precipitation
Radiation Solar radiation
Radiation_2Mon 2-month solar radiation
Tmax Maximum temperature
Tmax_2Mon 2-month maximum temperature
Tmin Minimum temperature
Tmin_2Mon 2-month minimum temperature
VP Vapor pressure
VP_2Mon 2-month vapor pressure

Testing datasets: Same structure as corresponding training datasets.

Details

Dataset Structure:

- **10var datasets**: Basic environmental variables (12 columns)
- **22var datasets**: Extended variables with climate indices (24 columns)
- **Training datasets**: Used for model building
- **Testing datasets**: Used for model evaluation

Climate Indices: IPO (Interdecadal Pacific Oscillation), Nino3.4 (El Nino), PDO (Pacific Decadal Oscillation), PNA (Pacific North American pattern)

Data Sources: ERA5 Land, Daymet, USGS, and climate indices databases

Source

Environmental monitoring stations, climate indices databases, ERA5 Land, Daymet, and USGS

---

summary *Summary methods for SCE and SCA models*

Description

Provide concise summaries of model structure and performance for SCE and SCA objects.

Usage

```
## S3 method for class 'sce'
summary(object, ...)
## S3 method for class 'sca'
summary(object, ...)
```

Arguments

`object`	An SCE or SCA model object
`...`	Additional arguments passed to or ignored by methods

Details

For `summary.sce`, the method prints ensemble configuration, out-of-bag (OOB) performance statistics, tree structure information, and tree weight distribution.\ For `summary.sca`, the method prints tree structure information and variable summaries for the single SCA tree.

Value

Invisibly returns the input `object` after printing the summary.

See Also

`sce`, `sca`, `print`, `importance`, `evaluate`

Index

air_quality_datasets, 2
air_quality_testing
        *(*air_quality_datasets*)*, 2
air_quality_training
        *(*air_quality_datasets*)*, 2

evaluate, 3, *6*, *8, 9*, *12*

importance, 4, *7–9*, *12*

Plot_RFE *(*plot_rfe*)*, 4
plot_rfe, 4, *7*
predict, *3*, 5, *8, 9*
print, 6, *12*

RFE_SCE *(*rfe_sce*)*, 7
rfe_sce, *4, 5*, 7

SCA *(*sca*)*, 8
sca, *3, 4*, *6*, 8, *9*, *12*
SCE *(*sce*)*, 9
sce, *3, 4*, *6–8*, 9, *12*
streamflow_datasets, 10
streamflow_testing_10var
        *(*streamflow_datasets*)*, 10
streamflow_testing_22var
        *(*streamflow_datasets*)*, 10
streamflow_training_10var
        *(*streamflow_datasets*)*, 10
streamflow_training_22var
        *(*streamflow_datasets*)*, 10
summary, *6*, 12