Package 'spfcICOMP'

September 3, 2026

Title Shrinkage Principal Fitted Components with Information Complexity-Based Model Selection

Version 0.1.0

Description Implements shrinkage principal fitted components for sufficient dimension reduction in high-dimensional regression and classification. Provides regularised covariance estimation using Oracle Approximating Shrinkage and Maximum Entropy Covariance, structural-dimension selection using conventional and information-complexity criteria, response-guided feature screening, reduced-space prediction, and simulation utilities. Methodological foundations include Cook and Forzani (2008) <doi:10.1214/08-STS275>, Chen et al. (2010) <doi:10.1109/TSP.2010.2053029>, Bozdogan (2000) <doi:10.1006/jmps.1999.1277>, and Olorede and Yahya (2019) <doi:10.48550/arXiv.1909.13017>.

License MIT + file LICENSE

URL https://github.com/ilovemaths/spfcICOMP

BugReports https://github.com/ilovemaths/spfcICOMP/issues

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Imports MASS

Suggests testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

Depends R (>= 3.5.0)

NeedsCompilation no

Author Kabir Opeyemi Olorede [aut, cre, cph]

Maintainer Kabir Opeyemi Olorede <kabir.olorede@kwasu.edu.ng>

Repository CRAN

Date/Publication 2026-09-03 13:50:02 UTC

Contents

`benchmark_spfc` *Benchmark SPFC Across Covariance Estimators*

Description

Fits SPFC models using several covariance estimators and compares their downstream predictive performance, runtime, shrinkage behaviour, and variable-selection output.

Usage

```
benchmark_spfc(
  X,
  y,
  d,
  ytype = c("auto", "continuous", "categorical"),
  methods = c("mec", "oas", "sre", "sde", "cse"),
  variable_method = "adaptive_weighted_l1",
  quantile_cut = 0.75,
  validation = c("resubstitution", "holdout", "cv"),
  test_fraction = 0.3,
  folds = 5,
  validation_seed = 123,
  verbose = TRUE,
  ...
)
```

Arguments

`X`	Numeric predictor matrix.
`y`	Response vector.
`d`	Structural dimension.
`ytype`	Character. One of `"auto"`, `"continuous"`, or `"categorical"`.
`methods`	Character vector of covariance estimators to compare.
`variable_method`	Variable-selection method.
`quantile_cut`	Quantile threshold for adaptive variable selection.

validation	Character. Validation strategy. One of "resubstitution", "holdout", or "cv".
test_fraction	Proportion of observations allocated to the test set when validation = "holdout".
folds	Number of folds used when validation = "cv".
validation_seed	Random seed used for train-test splitting or cross-validation fold creation.
verbose	Logical. If TRUE, print progress messages.
...	Additional arguments passed to spfc().

Value

A list of class spfc_benchmark.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
bench <- benchmark_spfc(
  X, y, d = 1, ytype = 'continuous',
  methods = c('oas', 'sre'),
  variable_method = 'adaptive_weighted_l1',
  quantile_cut = 0.75, validation = 'resubstitution',
  verbose = FALSE
)
bench$summary
```

---

binary_classification_metrics

Compute Binary Classification Metrics

Description

Computes confusion matrix, accuracy, sensitivity, specificity, precision, and F1 score for binary classification.

Usage

```
binary_classification_metrics(observed, predicted, positive = NULL)
```

Arguments

observed	Observed binary response.
predicted	Predicted binary class labels.
positive	Optional positive class label. If NULL, the second factor level is treated as the positive class.

Value

A list containing the confusion matrix and classification metrics.

Examples

```
obs <- factor(c('A', 'A', 'B', 'B'))
pred <- factor(c('A', 'B', 'B', 'B'), levels = levels(obs))
binary_classification_metrics(obs, pred, positive = 'B')
```

---

coef.spfc *Extract SPFC Fit from Complete Workflow*

Description

Extract SPFC Fit from Complete Workflow

Usage

```
## S3 method for class 'spfc'
coef(object, ...)
```

Arguments

object	Object returned by spfc().
...	Additional arguments.

Value

Estimated SPFC direction matrix from the fitted SPFC component.

---

coef.spfc_fit *Extract SPFC Coefficients*

Description

Extracts the estimated SPFC direction matrix.

Usage

```
## S3 method for class 'spfc_fit'
coef(object, ...)
```

Arguments

object	Object returned by spfc_fit().
...	Additional arguments.

Value

Numeric matrix of estimated SPFC directions.

`compare_covariance_methods` *Compare Covariance Estimators in SPFC*

Description

Fits shrinkage principal fitted components using several covariance estimators and evaluates the reduced model fitted on the resulting SPFC scores.

Usage

```
compare_covariance_methods(
  X,
  y,
  d,
  methods = c("mec", "oas", "sre", "sde", "cse"),
  ytype = c("auto", "continuous", "categorical"),
  nslices = 5,
  poly_degree = 1,
  gamma = 0.1,
  rho = 0.5,
  centre_x = TRUE,
  scale_x = FALSE,
  classifier = c("auto", "glm", "ridge_logistic"),
  eps = 1e-08,
  verbose = TRUE
)
```

Arguments

`X`	Numeric predictor matrix.
`y`	Response vector.
`d`	Number of reduction directions.
`methods`	Character vector of covariance estimators. Available options are `"mec"`, `"oas"`, `"sre"`, `"sde"`, and `"cse"`.
`ytype`	Character. One of `"auto"`, `"continuous"`, or `"categorical"`.
`nslices`	Number of slices for continuous responses.
`poly_degree`	Polynomial degree for continuous response basis.
`gamma`	Ridge constant for SRE.
`rho`	Convex shrinkage intensity for CSE.
`centre_x`	Logical. If `TRUE`, centre predictor columns.

scale_x	Logical. If TRUE, scale predictor columns.
classifier	Character. One of "auto", "glm", or "ridge_logistic".
eps	Positive numerical floor for eigenvalues.
verbose	Logical. If TRUE, print progress.

Value

A list containing a summary table and full fitted objects.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
cmp <- compare_covariance_methods(
  X, y, d = 1, methods = c('oas', 'sre'),
  ytype = 'continuous', poly_degree = 2, verbose = FALSE
)
cmp$summary
```

---

create_cv_folds *Create K-fold Cross-validation Folds*

Description

Create K-fold Cross-validation Folds

Usage

```
create_cv_folds(y, folds = 5, seed = 123, stratify = TRUE)
```

Arguments

y	Response vector.
folds	Number of folds.
seed	Random seed.
stratify	Logical. If TRUE, stratify factor, character, logical, or numeric two-class responses.

Value

Integer vector of fold labels.

Examples

```
y <- seq_len(12)
create_cv_folds(y, folds = 3, seed = 123, stratify = FALSE)
```

`create_spfc_design_grid` *Create an SPFC Simulation Design Grid*

Description

Creates a factorial simulation design grid for SPFC simulation studies.

Usage

```
create_spfc_design_grid(
  response_type = c("continuous", "categorical"),
  n_values = c(50, 100),
  p_values = c(20, 50),
  d_values = c(1, 2),
  s_values = c(5),
  rho_x_values = c(0.2, 0.5, 0.8),
  snr_values = c(1, 2),
  signal_strength_values = c(0.8, 1.2),
  cov_methods = c("mec", "oas", "sre", "sde", "cse"),
  variable_methods = c("adaptive_weighted_l1", "c1f_extension"),
  nrep = 10
)
```

Arguments

`response_type`	Character vector. One or both of `"continuous"` and `"categorical"`.
`n_values`	Sample sizes.
`p_values`	Predictor dimensions.
`d_values`	True structural dimensions.
`s_values`	Sparsity levels.
`rho_x_values`	Predictor correlation values.
`snr_values`	Signal-to-noise ratios for continuous responses.
`signal_strength_values`	Signal strengths for categorical responses.
`cov_methods`	Covariance estimators.
`variable_methods`	Variable-selection methods.
`nrep`	Number of Monte Carlo replicates per design setting.

Value

A data frame containing the full simulation design.

Examples

```
create_spfc_design_grid(
  response_type = 'continuous', n_values = 30, p_values = 8,
  d_values = 1, s_values = 3, rho_x_values = 0.3,
  snr_values = 2, cov_methods = 'oas',
  variable_methods = 'adaptive_weighted_l1', nrep = 1
)
```

`create_train_test_split` *Create Train-test Split*

Description

Create Train-test Split

Usage

```
create_train_test_split(y, test_fraction = 0.3, seed = 123, stratify = TRUE)
```

Arguments

`y`	Response vector.
`test_fraction`	Proportion assigned to test set.
`seed`	Random seed.
`stratify`	Logical. If TRUE, stratify factor, character, logical, or numeric two-class responses.

Value

A list with train and test indices.

Examples

```
y <- seq_len(20)
create_train_test_split(
  y, test_fraction = 0.25, seed = 123, stratify = FALSE
)
```

cse *Convex Sum Covariance Estimator*

Description

Computes a convex combination of the sample covariance matrix and a spherical target matrix.

Usage

```
cse(x, rho = 0.5)
```

Arguments

x	Numeric predictor matrix.
rho	Convex shrinkage intensity between 0 and 1.

Value

A list with covariance matrix covx, shrinkage intensity rho, and method label.

Examples

```
set.seed(1)
X <- matrix(rnorm(120), nrow = 20, ncol = 6)
est <- cse(X, rho = 0.5)
dim(est$covx)
```

estimate_covariance *Estimate a Regularised Covariance Matrix*

Description

Unified interface for covariance estimators used in shrinkage principal fitted components regression.

Usage

```
estimate_covariance(
  X,
  y = NULL,
  ytype = c("auto", "continuous", "categorical"),
  method = c("mec", "oas", "sre", "sde", "cse"),
  nslices = 5,
  gamma = 0.1,
  rho = 0.5,
  eps = 1e-08
)
```

Arguments

X	Numeric predictor matrix.
y	Optional response vector. Required when method = "mec".
ytype	Character. One of "auto", "continuous", or "categorical".
method	Covariance estimator. One of "mec", "oas", "sre", "sde", or "cse".
nslices	Number of slices for MEC with continuous responses.
gamma	Ridge constant for SRE.
rho	Convex shrinkage intensity for CSE.
eps	Positive numerical floor for eigenvalues.

Value

A list containing the covariance estimate, shrinkage intensity, and method diagnostics.

Examples

```
set.seed(1)
X <- matrix(rnorm(120), nrow = 20, ncol = 6)
est <- estimate_covariance(X, method = 'oas')
c(method = est$method, rho = est$rho)
```

---

evaluate_reduced_model

Evaluate a Reduced Model

Description

Evaluates a model fitted on reduced SPFC scores.

Usage

```
evaluate_reduced_model(object, Z, y, threshold = 0.5, positive = NULL)
```

Arguments

object	Object returned by fit_reduced_model().
Z	Numeric reduced-score matrix.
y	Observed response.
threshold	Classification threshold for binary responses.
positive	Optional positive class label.

Value

A list of evaluation metrics.

Examples

```
set.seed(1)
Z <- matrix(rnorm(40), nrow = 20, ncol = 2)
y <- Z[, 1] - 0.5 * Z[, 2] + rnorm(20, sd = 0.4)
mod <- fit_reduced_model(Z, y, ytype = 'continuous')
evaluate_reduced_model(mod, Z, y)
```

`evaluate_spfc_cv` *Evaluate SPFC with K-fold Cross-validation*

Description

Evaluate SPFC with K-fold Cross-validation

Usage

```
evaluate_spfc_cv(
  X,
  y,
  d,
  ytype = c("auto", "continuous", "categorical"),
  cov_method = c("mec", "oas", "sre", "sde", "cse"),
  folds = 5,
  seed = 123,
  nslices = 5,
  poly_degree = 1,
  variable_method = c("adaptive_weighted_l1", "c1f_extension"),
  selection_rule = c("c1f", "quantile", "fixed"),
  threshold = NULL,
  quantile_cut = 0.75,
  centre_x = TRUE,
  scale_x = FALSE,
  ...
)
```

Arguments

`X`	Predictor matrix.
`y`	Response vector.
`d`	Structural dimension.
`ytype`	Response type.
`cov_method`	Covariance estimator.
`folds`	Number of folds.
`seed`	Random seed.
`nslices`	Number of response slices.

poly_degree	Polynomial degree.
variable_method	Variable selection method.
selection_rule	Character screening rule: "c1f", "quantile", or "fixed".
threshold	Optional finite numeric screening threshold. It is required for selection_rule = "fixed"; for "quantile", NULL uses quantile_cut; it is ignored by the C1F-calibrated rule.
quantile_cut	Variable selection threshold.
centre_x	Logical.
scale_x	Logical.
...	Additional arguments passed to spfc_fit().

Value

A list containing fold-level and aggregate validation results.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
cv <- evaluate_spfc_cv(
  X, y, d = 1, ytype = 'continuous', cov_method = 'oas',
  folds = 3, seed = 123, poly_degree = 2,
  selection_rule = 'quantile', quantile_cut = 0.75
)
cv$aggregate
```

---

evaluate_spfc_train_test

Evaluate SPFC with Train-test Validation

Description

Evaluate SPFC with Train-test Validation

Usage

```
evaluate_spfc_train_test(
  X,
  y,
  train_idx,
  test_idx,
  d,
  ytype = c("auto", "continuous", "categorical"),
```

```
  cov_method = c("mec", "oas", "sre", "sde", "cse"),
  nslices = 5,
  poly_degree = 1,
  variable_method = c("adaptive_weighted_l1", "c1f_extension"),
  selection_rule = c("c1f", "quantile", "fixed"),
  threshold = NULL,
  quantile_cut = 0.75,
  centre_x = TRUE,
  scale_x = FALSE,
  ...
)
```

Arguments

X	Predictor matrix.
y	Response vector.
train_idx	Training indices.
test_idx	Test indices.
d	Structural dimension.
ytype	Response type.
cov_method	Covariance estimator.
nslices	Number of response slices.
poly_degree	Polynomial degree.
variable_method	Variable selection method.
selection_rule	Character screening rule: "c1f", "quantile", or "fixed".
threshold	Optional finite numeric screening threshold. It is required for selection_rule = "fixed"; for "quantile", NULL uses quantile_cut; it is ignored by the C1F-calibrated rule.
quantile_cut	Variable selection threshold.
centre_x	Logical.
scale_x	Logical.
...	Additional arguments passed to spfc_fit().

Value

A list containing fit, reduced model, predictions, and test metrics.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
split <- create_train_test_split(
  y, test_fraction = 0.25, seed = 123, stratify = FALSE
```

```
)
ans <- evaluate_spfc_train_test(
  X, y, train_idx = split$train, test_idx = split$test,
  d = 1, ytype = 'continuous', cov_method = 'oas',
  poly_degree = 2, selection_rule = 'quantile',
  quantile_cut = 0.75
)
ans$evaluation
```

`fitted.spfc` *Extract Reduced Scores from Complete Workflow*

Description

Extract Reduced Scores from Complete Workflow

Usage

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

Arguments

`object`	Object returned by `spfc()`.
`...`	Additional arguments.

Value

Reduced SPFC score matrix.

`fitted.spfc_fit` *Fitted SPFC Scores*

Description

Extracts the reduced SPFC score matrix.

Usage

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

Arguments

`object`	Object returned by `spfc_fit()`.
`...`	Additional arguments.

Value

Numeric matrix of reduced SPFC scores.

`fit_reduced_model` *Fit Reduced Downstream Model*

Description

Fits a downstream predictive model using the reduced SPFC score matrix.

Usage

```
fit_reduced_model(
  Z,
  y,
  ytype = c("auto", "continuous", "categorical"),
  classifier = c("auto", "glm", "ridge_logistic"),
  standardize = TRUE
)
```

Arguments

`Z`	Numeric reduced predictor matrix.
`y`	Response vector.
`ytype`	Character. One of `"auto"`, `"continuous"`, or `"categorical"`.
`classifier`	Character. Classification model. One of `"auto"`, `"glm"`, or `"ridge_logistic"`.
`standardize`	Logical. If `TRUE`, standardise predictors internally in ridge logistic regression.

Value

A list containing the fitted downstream model and metadata.

Examples

```
set.seed(1)
Z <- matrix(rnorm(40), nrow = 20, ncol = 2)
y <- Z[, 1] - 0.5 * Z[, 2] + rnorm(20, sd = 0.4)
fit_reduced_model(Z, y, ytype = 'continuous')
```

mec *Maximum Entropy Covariance Estimator*

Description

Computes the Maximum Entropy Covariance estimator for sufficient dimension reduction. The response is first sliced. Slice-specific covariance matrices are computed, the entropy-maximising slice covariance is selected, and an entropy-guided covariance estimate is stabilised by a convex shrinkage step.

Usage

```
mec(
  x,
  y,
  ytype = c("auto", "continuous", "categorical"),
  nslices = 5,
  eps = 1e-08
)
```

Arguments

x	Numeric predictor matrix.
y	Response vector.
ytype	Character. One of "auto", "continuous", or "categorical".
nslices	Number of slices for continuous responses.
eps	Positive numerical floor for eigenvalues.

Value

A list with covariance matrix covx, shrinkage intensity rho, method label, response type, number of slices used, and entropy values.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
est <- mec(X, y, ytype = 'continuous', nslices = 5)
c(method = est$method, rho = est$rho)
```

oas *Oracle Approximating Shrinkage-Type Covariance Estimator*

Description

Computes an OAS-type stabilised covariance estimator with an additional convex shrinkage step.

Usage

```
oas(x)
```

Arguments

x	Numeric predictor matrix.

Value

A list with covariance matrix covx, shrinkage intensity rho, and method label.

Examples

```
set.seed(1)
X <- matrix(rnorm(120), nrow = 20, ncol = 6)
est <- oas(X)
c(method = est$method, rho = est$rho)
```

plot_dimension_selection *Plot Dimension-Selection Criteria*

Description

Plots information criteria returned by spfc_select_dimension().

Usage

```
plot_dimension_selection(
  dsel,
  criterion = c("AIC", "BIC", "CAIC", "ICOMP_IFIM", "ICOMP_MISSPEC", "CICOMP")
)
```

Arguments

dsel	Object returned by spfc_select_dimension().
criterion	Character. Criterion to plot.

Value

Invisibly returns the plotted data.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
dsel <- spfc_select_dimension(
  X, y, d_grid = 1:2, cov_method = 'oas',
  ytype = 'continuous', poly_degree = 2, verbose = FALSE
)
plot_dimension_selection(dsel, criterion = 'ICOMP_IFIM')
```

---

plot_metric_by_covariance

Plot SPFC Simulation Metric by Covariance Method

Description

Produces a base R boxplot for a selected simulation metric across covariance estimators.

Usage

```
plot_metric_by_covariance(results, metric, main = NULL, ylab = NULL)
```

Arguments

`results`	Data frame returned by `run_spfc_design()` or `run_spfc_simulation()`.
`metric`	Character. Name of the metric column to plot.
`main`	Optional plot title.
`ylab`	Optional y-axis label.

Value

Invisibly returns the formula used for plotting.

Examples

```
results <- data.frame(
  cov_method = rep(c('oas', 'sre'), each = 2),
  rmse = c(0.50, 0.55, 0.62, 0.59),
  runtime_sec = c(0.01, 0.012, 0.008, 0.009),
  subspace_distance = c(0.20, 0.22, 0.28, 0.25)
)
plot_metric_by_covariance(results, metric = 'rmse')
```

`plot_rmse_by_covariance` *Plot RMSE by Covariance Method*

Description

Plot RMSE by Covariance Method

Usage

```
plot_rmse_by_covariance(results)
```

Arguments

`results`	Simulation results data frame.

Value

Invisibly returns the plotting formula.

Examples

```
results <- data.frame(
  cov_method = rep(c('oas', 'sre'), each = 2),
  rmse = c(0.50, 0.55, 0.62, 0.59),
  runtime_sec = c(0.01, 0.012, 0.008, 0.009),
  subspace_distance = c(0.20, 0.22, 0.28, 0.25)
)
plot_rmse_by_covariance(results)
```

`plot_runtime_by_covariance` *Plot Runtime by Covariance Method*

Description

Plot Runtime by Covariance Method

Usage

```
plot_runtime_by_covariance(results)
```

Arguments

`results`	Simulation results data frame.

Value

Invisibly returns the plotting formula.

Examples

```
results <- data.frame(
  cov_method = rep(c('oas', 'sre'), each = 2),
  rmse = c(0.50, 0.55, 0.62, 0.59),
  runtime_sec = c(0.01, 0.012, 0.008, 0.009),
  subspace_distance = c(0.20, 0.22, 0.28, 0.25)
)
plot_runtime_by_covariance(results)
```

---

plot_spfc_scores *Plot SPFC Reduced Scores*

Description

Produces a reduced-score plot from an object returned by `spfc_fit()`.

Usage

```
plot_spfc_scores(fit, y = NULL, dims = c(1, 2), main = NULL)
```

Arguments

`fit`	Object returned by `spfc_fit()`.
`y`	Optional response vector used for colouring or grouping.
`dims`	Integer vector of length 1 or 2 indicating SPFC score dimensions.
`main`	Optional plot title.

Value

Invisibly returns the plotted data.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
fit <- spfc_fit(X, y, d = 1, ytype = 'continuous',
                cov_method = 'oas', poly_degree = 2)
plot_spfc_scores(fit, y = y, dims = 1)
```

plot_subspace_distance_by_covariance

Plot Subspace Distance by Covariance Method

Description

Plot Subspace Distance by Covariance Method

Usage

```
plot_subspace_distance_by_covariance(results)
```

Arguments

`results`	Simulation results data frame.

Value

Invisibly returns the plotting formula.

Examples

```
results <- data.frame(
  cov_method = rep(c('oas', 'sre'), each = 2),
  rmse = c(0.50, 0.55, 0.62, 0.59),
  runtime_sec = c(0.01, 0.012, 0.008, 0.009),
  subspace_distance = c(0.20, 0.22, 0.28, 0.25)
)
plot_subspace_distance_by_covariance(results)
```

predict.ridge_logistic_fit

Predict from Ridge Logistic Regression

Description

Predict from Ridge Logistic Regression

Usage

```
## S3 method for class 'ridge_logistic_fit'
predict(object, newdata, type = c("class", "response"), ...)
```

Arguments

object	Object returned by fit_ridge_logistic().
newdata	Numeric reduced predictor matrix.
type	Character. One of "class" or "response".
...	Additional arguments.

Value

Predicted classes or probabilities.

---

predict.spfc *Predict SPFC Scores from Complete Workflow*

Description

Predict SPFC Scores from Complete Workflow

Usage

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

Arguments

object	Object returned by spfc().
newdata	Numeric matrix or data frame of new predictor observations.
...	Additional arguments.

Value

Numeric matrix of predicted SPFC scores.

---

predict.spfc_fit *Predict SPFC Scores for New Data*

Description

Projects new predictor observations into the estimated SPFC reduced space.

Usage

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

Arguments

object	Object returned by spfc_fit().
newdata	Numeric matrix or data frame of new predictor observations.
...	Additional arguments.

Value

Numeric matrix of predicted SPFC scores.

---

predict_reduced_model *Predict from a Reduced Model Fit*

Description

Generates fitted values, probabilities, or predicted classes from an object returned by fit_reduced_model().

Usage

```
predict_reduced_model(
  object,
  Z,
  type = c("response", "class", "link"),
  threshold = 0.5
)
```

Arguments

object	Object returned by fit_reduced_model().
Z	Numeric reduced-score matrix.
type	Character. One of "response", "class", or "link".
threshold	Classification threshold for binary probabilities.

Value

Numeric predictions or factor class labels.

Examples

```
set.seed(1)
Z <- matrix(rnorm(40), nrow = 20, ncol = 2)
y <- Z[, 1] - 0.5 * Z[, 2] + rnorm(20, sd = 0.4)
mod <- fit_reduced_model(Z, y, ytype = 'continuous')
head(predict_reduced_model(mod, Z, type = 'response'))
```

print.spfc *Print Complete SPFC Workflow*

Description

Print Complete SPFC Workflow

Usage

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

Arguments

x	Object returned by spfc().
...	Additional arguments.

Value

Invisibly returns x.

print.spfc_benchmark *Print SPFC Benchmark*

Description

Print SPFC Benchmark

Usage

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

Arguments

x	Object returned by benchmark_spfc().
...	Additional arguments.

Value

Invisibly returns x.

print.spfc_fit *Print an SPFC Fit*

Description

Print an SPFC Fit

Usage

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

Arguments

x	Object returned by spfc_fit().
...	Additional arguments.

Value

Invisibly returns x.

print.summary.spfc *Print Summary of Complete SPFC Workflow*

Description

Print Summary of Complete SPFC Workflow

Usage

```
## S3 method for class 'summary.spfc'
print(x, ...)
```

Arguments

x	Object returned by summary.spfc().
...	Additional arguments.

Value

Invisibly returns x.

print.summary.spfc_benchmark

Print Summary of SPFC Benchmark

Description

Print Summary of SPFC Benchmark

Usage

```
## S3 method for class 'summary.spfc_benchmark'
print(x, ...)
```

Arguments

x	Object returned by summary.spfc_benchmark().
...	Additional arguments.

Value

Invisibly returns x.

print.summary.spfc_fit

Print Summary of an SPFC Fit

Description

Print Summary of an SPFC Fit

Usage

```
## S3 method for class 'summary.spfc_fit'
print(x, ...)
```

Arguments

x	Object returned by summary.spfc_fit().
...	Additional arguments.

Value

Invisibly returns x.

rank_covariance_methods *Rank Covariance Methods from Simulation Summaries*

Description

Ranks covariance estimators according to a chosen performance metric.

Usage

```
rank_covariance_methods(summary, metric, smaller_is_better = TRUE)
```

Arguments

summary	Data frame returned by summarise_spfc_simulation().
metric	Metric column used for ranking.
smaller_is_better	Logical. If TRUE, lower metric values are better.

Value

Ranked data frame.

Examples

```
summary <- data.frame(
  cov_method = c('oas', 'sre'),
  mean_rmse = c(0.50, 0.60)
)
rank_covariance_methods(summary, metric = 'mean_rmse')
```

run_spfc_design *Run an SPFC Simulation Design*

Description

Runs an SPFC simulation design grid sequentially.

Usage

```
run_spfc_design(
  design,
  base_seed = 123,
  d_grid = 1:5,
  criteria = c("AIC", "BIC", "CAIC", "ICOMP_IFIM", "CICOMP"),
  selection_rule = "c1f",
  nslices = 5,
  poly_degree = 2,
  gamma = 0.1,
  rho = 0.5,
  save_every = NULL,
  save_path = "spfc_simulation_partial.rds",
  verbose = TRUE
)
```

Arguments

design	Data frame returned by create_spfc_design_grid().
base_seed	Integer base seed.
d_grid	Candidate structural dimensions evaluated for each simulation design row.
criteria	Character vector of information criteria used for structural-dimension selection.
selection_rule	Character screening rule passed to run_spfc_design_row().
nslices	Number of slices.
poly_degree	Polynomial degree.
gamma	Ridge constant for SRE.
rho	Convex shrinkage intensity for CSE.
save_every	Optional integer. If supplied, intermediate results are saved every save_every rows.
save_path	Optional path for intermediate RDS output.
verbose	Logical. If TRUE, print progress.

Value

A data frame of simulation results.

Examples

```
design <- create_spfc_design_grid(
  response_type = 'continuous', n_values = 30, p_values = 8,
  d_values = 1, s_values = 3, rho_x_values = 0.3,
  snr_values = 2, cov_methods = 'oas',
  variable_methods = 'adaptive_weighted_l1', nrep = 1
)
ans <- run_spfc_design(
  design, base_seed = 123, d_grid = 1, criteria = 'AIC',
```

```
  selection_rule = 'c1f', poly_degree = 2, verbose = FALSE
)
ans[, c('cov_method', 'criterion', 'fitted_d', 'rmse')]
```

`run_spfc_simulation` *Run a Small SPFC Simulation Study*

Description

Run a Small SPFC Simulation Study

Usage

```
run_spfc_simulation(
  response_type = c("continuous", "categorical"),
  nrep = 10,
  n = 100,
  p = 50,
  d = 1,
  s = 5,
  rho_x = 0.5,
  snr = 1,
  signal_strength = 1,
  cov_methods = c("mec", "oas", "sre", "sde", "cse"),
  variable_methods = c("adaptive_weighted_l1", "c1f_extension"),
  seed = 123
)
```

Arguments

`response_type`	Character. `"continuous"` or `"categorical"`.
`nrep`	Number of Monte Carlo replicates.
`n`	Sample size.
`p`	Number of predictors.
`d`	Structural dimension.
`s`	Number of active variables.
`rho_x`	Predictor correlation.
`snr`	Signal-to-noise ratio for continuous response.
`signal_strength`	Signal strength for categorical response.
`cov_methods`	Covariance estimators to compare.
`variable_methods`	Variable-selection methods to compare.
`seed`	Random seed.

Value

A data frame of simulation results.

Examples

```
ans <- run_spfc_simulation(
  response_type = 'continuous', nrep = 1, n = 30, p = 8,
  d = 1, s = 3, rho_x = 0.3, snr = 2,
  cov_methods = c('oas', 'sre'),
  variable_methods = 'adaptive_weighted_l1', seed = 123
)
ans[, c('cov_method', 'rmse', 'subspace_distance')]
```

---

score_information_criteria

Score Information Criteria for a Reduced Model

Description

Computes AIC, BIC, CAIC, ICOMP(IFIM), ICOMP(Misspec), and CICOMP for a model fitted on reduced SPFC scores.

Usage

```
score_information_criteria(reduced_model, n, eps = 1e-10)
```

Arguments

`reduced_model`	Object returned by `fit_reduced_model()`.
`n`	Sample size.
`eps`	Small positive numerical floor.

Value

A one-row data frame of information criteria.

Examples

```
set.seed(1)
Z <- matrix(rnorm(40), nrow = 20, ncol = 2)
y <- Z[, 1] - 0.5 * Z[, 2] + rnorm(20, sd = 0.4)
mod <- fit_reduced_model(Z, y, ytype = 'continuous')
score_information_criteria(mod, n = length(y))
```

sde *Stipulated Diagonal Covariance Estimator*

Description

Computes a diagonal covariance estimator by retaining only the diagonal elements of the sample covariance matrix.

Usage

```
sde(x)
```

Arguments

x	Numeric predictor matrix.

Value

A list with covariance matrix `covx`, shrinkage intensity `rho`, and method label.

Examples

```
set.seed(1)
X <- matrix(rnorm(120), nrow = 20, ncol = 6)
est <- sde(X)
c(method = est$method, rho = est$rho)
```

simulate_spfc_binary *Simulate Binary High-Dimensional Classification Data*

Description

Simulate Binary High-Dimensional Classification Data

Usage

```
simulate_spfc_binary(
  n = 100,
  p = 50,
  d = 1,
  s = 5,
  rho_x = 0.5,
  signal_strength = 1,
  seed = NULL
)
```

Arguments

n	Sample size.
p	Number of predictors.
d	Structural dimension.
s	Number of active variables.
rho_x	Predictor correlation.
signal_strength	Signal strength in the logistic model.
seed	Optional random seed.

Value

A list containing X, y, true directions, active set, and covariance.

Examples

```
sim <- simulate_spfc_binary(
  n = 30, p = 8, d = 1, s = 3, rho_x = 0.3,
  signal_strength = 1, seed = 123
)
table(sim$y)
```

---

simulate_spfc_continuous

Simulate Continuous High-Dimensional PFC Data

Description

Generates data directly from a principal fitted components inverse model, so the nominal structural dimension is identifiable by construction. For `d = 1`, the inverse mean uses `f(Y) = Y`. For `d = 2`, it uses the two population-orthonormal basis functions `Y` and `(Y^2 - 1)/sqrt(2)` with `Y ~ N(0,1)`. The inverse-mean loading matrix is constructed so that the true sufficient reduction subspace is the sparse matrix returned as `B_true`, even when the residual predictor covariance is correlated.

Usage

```
simulate_spfc_continuous(
  n = 100,
  p = 50,
  d = 1,
  s = 5,
  rho_x = 0.5,
  snr = 1,
  n_test = n,
  seed = NULL
)
```

Arguments

n	Training sample size.
p	Number of predictors.
d	Structural dimension. The validated thesis DGP supports 1 or 2.
s	Number of active variables.
rho_x	AR(1) correlation parameter of the inverse-model residual covariance.
snr	Inverse-model signal-to-noise ratio in the sufficient coordinates.
n_test	Independent test-sample size. Defaults to n.
seed	Optional random seed.

Details

An independent test sample is generated from the same inverse model for honest out-of-sample prediction assessment.

Value

A list containing training and test data, true PFC quantities, sparse reduction directions, active set, and residual covariance.

Examples

```
sim <- simulate_spfc_continuous(
  n = 30, p = 8, d = 1, s = 3, rho_x = 0.3,
  snr = 2, n_test = 10, seed = 123
)
c(train_n = length(sim$y), test_n = length(sim$y_test))
```

---

spfc *Fit a Complete SPFC Workflow*

Description

Fits a complete Shrinkage Principal Fitted Components workflow, including optional structural dimension selection, SPFC fitting, reduced model fitting, evaluation, and variable selection.

Usage

```
spfc(
  X,
  y,
  d = NULL,
  d_grid = 1:3,
  ytype = c("auto", "continuous", "categorical"),
  cov_method = c("mec", "oas", "sre", "sde", "cse"),
```

```
  selector = c("AIC", "BIC", "CAIC", "ICOMP_IFIM", "ICOMP_MISSPEC", "CICOMP"),
  nslices = 5,
  poly_degree = 1,
  variable_method = c("adaptive_weighted_l1", "c1f_extension"),
  selection_rule = c("c1f", "quantile", "fixed"),
  threshold = NULL,
  quantile_cut = 0.75,
  validation = c("resubstitution", "holdout", "cv"),
  test_fraction = 0.3,
  folds = 5,
  validation_seed = 123,
  centre_x = TRUE,
  scale_x = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

X	Numeric predictor matrix.
y	Response vector.
d	Optional structural dimension. If NULL, dimension selection is performed.
d_grid	Candidate structural dimensions used when d = NULL.
ytype	Character. One of "auto", "continuous", or "categorical".
cov_method	Covariance estimator. One of "mec", "oas", "sre", "sde", or "cse".
selector	Criterion for selecting structural dimension when d = NULL.
nslices	Number of slices for continuous responses.
poly_degree	Polynomial degree for continuous response basis.
variable_method	Variable selection method.
selection_rule	Character screening rule: "c1f", "quantile", or "fixed".
threshold	Optional finite numeric screening threshold. It is required for selection_rule = "fixed"; for "quantile", NULL uses quantile_cut; it is ignored by the C1F-calibrated rule.
quantile_cut	Quantile threshold for variable selection.
validation	Character. Validation strategy. One of "resubstitution", "holdout", or "cv".
test_fraction	Numeric value between 0 and 1 giving the test-set proportion when validation = "holdout".
folds	Integer number of folds to use when validation = "cv".
validation_seed	Integer seed used for train-test splitting or cross-validation fold creation.
centre_x	Logical. If TRUE, centre predictor columns.
scale_x	Logical. If TRUE, scale predictor columns.
verbose	Logical. If TRUE, print progress messages.
...	Additional arguments passed to spfc_fit().

Value

A list of class `spfc` containing the complete workflow.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
obj <- spfc(
  X, y, d = 1, ytype = 'continuous', cov_method = 'oas',
  poly_degree = 2, selection_rule = 'quantile',
  quantile_cut = 0.75, validation = 'resubstitution',
  verbose = FALSE
)
obj
```

---

spfc_fit *Fit Shrinkage Principal Fitted Components*

Description

Fits a shrinkage principal fitted components model by replacing the unstable covariance object in the PFC reduction step with a selected shrinkage covariance estimator.

Usage

```
spfc_fit(
  X,
  y,
  d,
  fy = NULL,
  ytype = c("auto", "continuous", "categorical"),
  cov_method = c("mec", "oas", "sre", "sde", "cse"),
  nslices = 5,
  poly_degree = 1,
  gamma = 0.1,
  rho = 0.5,
  centre_x = TRUE,
  scale_x = FALSE,
  eps = 1e-08
)
```

Arguments

`X`	Numeric predictor matrix with observations in rows and variables in columns.
`y`	Response vector.

d	Number of reduction directions.
fy	Optional user-supplied response basis matrix.
ytype	Character. One of "auto", "continuous", or "categorical".
cov_method	Covariance estimator. One of "mec", "oas", "sre", "sde", or "cse".
nslices	Number of response slices for continuous responses.
poly_degree	Polynomial degree for continuous response basis construction.
gamma	Ridge constant for SRE.
rho	Convex shrinkage intensity for CSE.
centre_x	Logical. If TRUE, centre predictor columns.
scale_x	Logical. If TRUE, scale predictor columns.
eps	Positive numerical floor for eigenvalues.

Value

A spfc_fit object containing reduced scores, directions, fitted covariance objects, response basis, preprocessing metadata, and diagnostics.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
fit <- spfc_fit(
  X, y, d = 1, ytype = 'continuous',
  cov_method = 'oas', poly_degree = 2
)
coef(fit)
```

---

spfc_select_dimension *Select Structural Dimension for SPFC*

Description

Fits SPFC models across candidate dimensions and scores classical and information-complexity-based criteria.

Usage

```
spfc_select_dimension(
  X,
  y,
  d_grid = 1:3,
  cov_method = c("mec", "oas", "sre", "sde", "cse"),
  ytype = c("auto", "continuous", "categorical"),
  nslices = 5,
```

```
  poly_degree = 1,
  gamma = 0.1,
  rho = 0.5,
  centre_x = TRUE,
  scale_x = FALSE,
  classifier = c("auto", "glm"),
  eps = 1e-08,
  verbose = TRUE
)
```

Arguments

X	Numeric predictor matrix.
y	Response vector.
d_grid	Integer vector of candidate structural dimensions.
cov_method	Covariance estimator. One of "mec", "oas", "sre", "sde", or "cse".
ytype	Character. One of "auto", "continuous", or "categorical".
nslices	Number of slices for MEC with continuous responses.
poly_degree	Polynomial degree for continuous response basis.
gamma	Ridge constant for SRE.
rho	Convex shrinkage intensity for CSE.
centre_x	Logical. If TRUE, centre predictor columns.
scale_x	Logical. If TRUE, scale predictor columns.
classifier	Character. One of "auto" or "glm" for categorical responses when information criteria are required. Dimension selection uses an ordinary logistic companion model for categorical responses because the current information criteria require log-likelihood and covariance estimates.
eps	Small positive numerical floor.
verbose	Logical. If TRUE, print progress.

Value

A list containing the criterion table and selected dimensions by criterion.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
dsel <- spfc_select_dimension(
  X, y, d_grid = 1:2, cov_method = 'oas',
  ytype = 'continuous', poly_degree = 2, verbose = FALSE
)
dsel$selected
```

spfc_select_variables *Select Variables from SPFC Directions*

Description

Selects variables using either the thesis-safe adaptive weighted L1 route or the experimental C1F-informed extension route.

Usage

```
spfc_select_variables(
  fit,
  method = c("adaptive_weighted_l1", "c1f_extension"),
  selection_rule = c("c1f", "quantile", "fixed"),
  reduced_model = NULL,
  threshold = NULL,
  quantile_cut = 0.75,
  gamma = 1,
  c1f_calibration = c("icomp_hd_floor", "robust_universal", "raw"),
  eps = 1e-10
)
```

Arguments

fit	Object returned by spfc_fit().
method	Character. One of "adaptive_weighted_l1" or "c1f_extension".
selection_rule	Character. One of "c1f", "quantile", or "fixed".
reduced_model	Fitted reduced model used to obtain the model-parameter covariance when selection_rule = "c1f".
threshold	Numeric threshold applied to variable importance scores. If NULL, a quantile-based threshold is used.
quantile_cut	Numeric quantile used when threshold = NULL.
gamma	Positive adaptivity exponent for adaptive_weighted_l1.
c1f_calibration	Character. "icomp_hd_floor" is the frozen thesis rule that combines a robust empirical loading scale, the high-dimensional multiplicity floor sqrt(2 log(p) / n), and the bounded C1F multiplier 1 + C1F/(1 + C1F). "robust_universal" retains the version 0.0.0.9004 calibration for reproducibility, and "raw" retains the historical uncalibrated C1F route.
eps	Small positive numerical constant.

Value

A data frame containing variable indices, importance scores, weights, and selected status.

Examples

```
set.seed(1)
X <- matrix(rnorm(180), nrow = 30, ncol = 6)
y <- X[, 1] - 0.7 * X[, 2] + rnorm(30, sd = 0.5)
fit <- spfc_fit(X, y, d = 1, ytype = 'continuous',
                cov_method = 'oas', poly_degree = 2)
reduced_model <- fit_reduced_model(
  fit$Z, y, ytype = 'continuous'
)
sel <- spfc_select_variables(
  fit, method = 'c1f_extension', selection_rule = 'c1f',
  reduced_model = reduced_model
)
sel[, c('variable', 'shrunk_importance', 'selected')]
```

---

sre *Stipulated Ridge Covariance Estimator*

Description

Computes a ridge-stabilised covariance estimator by adding a positive constant to the diagonal of the sample covariance matrix.

Usage

```
sre(x, gamma = 0.1)
```

Arguments

x	Numeric predictor matrix.
gamma	Ridge constant.

Value

A list with covariance matrix covx, ridge intensity rho, and method label.

Examples

```
set.seed(1)
X <- matrix(rnorm(120), nrow = 20, ncol = 6)
est <- sre(X, gamma = 0.1)
c(method = est$method, rho = est$rho)
```

subspace_distance *Compute Subspace Distance*

Description

Computes a sine-angle distance between two subspaces.

Usage

```
subspace_distance(B_true, B_hat)
```

Arguments

B_true	True basis matrix.
B_hat	Estimated basis matrix.

Value

Numeric subspace distance.

Examples

```
B1 <- matrix(c(1, 0, 0), ncol = 1)
B2 <- matrix(c(0.9, 0.1, 0), ncol = 1)
subspace_distance(B1, B2)
```

summarise_spfc_simulation *Summarise SPFC Simulation Results*

Description

Aggregates Monte Carlo simulation results by covariance estimator, variable-selection method, and design factors.

Usage

```
summarise_spfc_simulation(results)
```

Arguments

results	Data frame returned by run_spfc_simulation().

Value

A summary data frame.

Examples

```
results <- run_spfc_simulation(
  response_type = 'continuous', nrep = 1, n = 30, p = 8,
  d = 1, s = 3, rho_x = 0.3, snr = 2,
  cov_methods = c('oas', 'sre'),
  variable_methods = 'adaptive_weighted_l1', seed = 123
)
summarise_spfc_simulation(results)
```

---

summary.spfc *Summarise Complete SPFC Workflow*

Description

Summarise Complete SPFC Workflow

Usage

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

Arguments

object	Object returned by spfc().
...	Additional arguments.

Value

A list of class summary.spfc.

---

summary.spfc_benchmark *Summarise SPFC Benchmark*

Description

Summarise SPFC Benchmark

Usage

```
## S3 method for class 'spfc_benchmark'
summary(object, metric = NULL, smaller_is_better = NULL, ...)
```

Arguments

object	Object returned by benchmark_spfc().
metric	Optional selection metric.
smaller_is_better	Optional logical indicating whether smaller values are better.
...	Additional arguments.

Value

A list of class summary.spfc_benchmark.

---

summary.spfc_fit *Summarise an SPFC Fit*

Description

Summarise an SPFC Fit

Usage

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

Arguments

object	Object returned by spfc_fit().
...	Additional arguments.

Value

A list containing key SPFC summary components.

---

variable_selection_metrics *Compute Variable-Selection Performance*

Description

Compute Variable-Selection Performance

Usage

```
variable_selection_metrics(selected, active_set, p)
```

Arguments

selected	Integer vector of selected variable indices.
active_set	Integer vector of truly active variable indices.
p	Total number of variables.

Value

A list of precision, recall, F1, false positives, and false negatives.

Examples

```
variable_selection_metrics(
  selected = c(1, 2, 5), active_set = c(1, 2, 3), p = 8
)
```

Index