

Package: dipca (via r-universe)

October 6, 2026

Type Package

Title Dynamic Inner Principal Component Analysis

Version 0.1.0

Description Methods for dynamic inner principal component analysis (DiPCA), dynamic inner canonical correlation analysis (DiCCA), and dynamic inner partial least squares (DiPLS) for multivariate time series. These methods extract latent components that follow autoregressive models, enabling temporal prediction and reconstruction. Based on Dong and Qin (2018) [<doi:10.1016/j.jprocont.2017.05.002>](https://doi.org/10.1016/j.jprocont.2017.05.002), Dong and Qin (2018) [<doi:10.1016/j.ifacol.2018.09.379>](https://doi.org/10.1016/j.ifacol.2018.09.379), and Dong and Qin (2018) [<doi:10.1016/j.jprocont.2018.04.006>](https://doi.org/10.1016/j.jprocont.2018.04.006). Integrates with the 'multivarious' package for consistent preprocessing and projection interfaces.

License GPL (>= 3) + file LICENSE

Encoding UTF-8

Language en-US

URL <https://github.com/bbuchtsbaum/dipca>

BugReports <https://github.com/bbuchtsbaum/dipca/issues>

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.0), multivarious (>= 0.3.0), stats, forecast

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), knitr, rmarkdown, ggplot2, gridExtra, reshape2

Config/testthat/edition 3

VignetteBuilder knitr

Repository <https://bbuchtsbaum.r-universe.dev>

Date/Publication 2026-06-25 00:25:13 UTC

RemoteUrl <https://github.com/bbuchtsbaum/dipca>

RemoteRef HEAD
RemoteSha d82339a377e41c41fff7954b9921c1bea2cb65f7

Contents

arima_dicca	2
dicca	3
dicca_predict_scores	6
dicca_scores	6
dipca	7
dipls	9
dipls_scores	10
predict.dicca	11
predict.dipca	12
predict.dipls	12
residuals.dicca	13
residuals.dipca	14
residuals.dipls	14

Index	16
--------------	-----------

arima_dicca	<i>Compact DiCCA with AR/ARMA/ARIMA inner models (measurement-space update)</i>
-------------	---

Description

Implements the measurement-space DiCCA alternating scheme with an inner AR/ARMA/ARIMA model fitted on each component. The deflation step follows Dong & Qin (2018): $X \leftarrow X - tp^\top$ with $p = X^\top t / (t^\top t)$.

Usage

```
arima_dicca(  
  X,  
  n_comp,  
  burnin = 0,  
  center = TRUE,  
  scale = FALSE,  
  inner = c("arma", "arima", "ar"),  
  order_mode = c("select_once", "fixed", "auto_each"),  
  max_iter = 200L,  
  max_iter_one = 100L,  
  tol = 1e-06,  
  order = NULL,  
  pmax = 5L,  
  qmax = 5L,  
)
```

```

    verbose = 1L,
    fit_every = Inf,
    experimental = FALSE
  )

```

Arguments

<code>X</code>	numeric matrix (T x m), rows are time, columns variables.
<code>n_comp</code>	number of dynamic latent variables to extract.
<code>burnin</code>	integer lag order s (≥ 1). Historically the compact routine used a burn-in; we treat this as the lag order for consistency with the main <code>dicca()</code> function.
<code>center</code>	logical; center columns before fitting.
<code>scale</code>	logical; scale to unit variance if TRUE.
<code>inner</code>	one of "ar", "arma", or "arima" (experimental; see experimental argument).
<code>order_mode</code>	how to handle ARIMA orders across iterations: "select_once" (default) selects once per component and reuses; "fixed" requires the order argument; "auto_each" re-selects at every iteration.
<code>max_iter</code>	integer maximum outer iterations for each component.
<code>max_iter_one</code>	retained for backwards compatibility; the effective iteration cap is $\min(\text{max_iter}, \text{max_iter_one})$.
<code>tol</code>	numeric convergence tolerance on the objective.
<code>order</code>	optional list(p=, d=, q=) when <code>order_mode = "fixed"</code> .
<code>pmax, qmax</code>	upper bounds when auto-selecting orders.
<code>verbose</code>	integer verbosity (≥ 0).
<code>fit_every</code>	integer; refit the inner model every <code>fit_every</code> iterations (default: Inf, fit only once).
<code>experimental</code>	logical; when TRUE enables the experimental <code>inner = "arima"</code> pathway, otherwise requesting "arima" will error unless the option <code>options(dipca.experimental_arima = TRUE)</code> has been set.

Value

a list with fields: W, P, T, models, info, R, method; with class 'dicca_model'.

<code>dicca</code>	<i>Dynamic-Inner Canonical Correlation Analysis (DiCCA) Extract dynamic latent variables by maximizing correlation between each latent series and its AR(s) prediction (canonical correlation form). This follows Dong & Qin (2018 IFAC) iteration and deflation.</i>
--------------------	---

Description

Dynamic-Inner Canonical Correlation Analysis (DiCCA) Extract dynamic latent variables by maximizing correlation between each latent series and its AR(s) prediction (canonical correlation form). This follows Dong & Qin (2018 IFAC) iteration and deflation.

Usage

```
dicca(
  X,
  s,
  l,
  preproc = multivarious::center(),
  tol = 1e-07,
  max_iter = 1000,
  n_init = 4,
  verbose = 0,
  seed = NULL,
  inner = c("classic", "ar", "arma", "arima")
)

DiCCA(
  X,
  s,
  l,
  preproc = multivarious::center(),
  tol = 1e-07,
  max_iter = 1000,
  n_init = 4,
  verbose = 0,
  seed = NULL,
  inner = c("classic", "ar", "arma", "arima")
)
```

Arguments

<code>X</code>	numeric matrix (T x m), time by variables.
<code>s</code>	integer lag order (≥ 1).
<code>l</code>	integer number of components to extract.
<code>preproc</code>	a <code>pre_processor</code> object from the <code>multivarious</code> package. Default is <code>center()</code> which centers columns. Use <code>prep(pass())</code> for no preprocessing.
<code>tol</code>	numeric outer tolerance ($\ w_{k+1} - w_{kl_inf}\ $).
<code>max_iter</code>	integer max outer iterations per component.
<code>n_init</code>	integer random restarts per component; best objective kept.
<code>verbose</code>	integer verbosity (0=quiet).
<code>seed</code>	optional RNG seed.

`inner` one of "classic" (DiCCA coordinate updates), or compact variants "ar", "arma", "arima" (the latter is experimental and requires `options(dipca.experimental_arima = TRUE)`) which dispatch to the compact ARIMA-DiCCA routine.

Details

One-component iteration: see DiCCA Eqs. (4)–(8): $\beta = (T_s^T T_s)^{-1} T_s^T t_{s+1}$; $\beta \leftarrow \beta / \sqrt{t_{s+1}^T T_s \beta}$; $X_{\beta} = \sum_i \beta_i X_{s+1-i}$; $w \leftarrow (X_{s+1}^T X_{s+1} + X_{\beta}^T X_{\beta})^{-1} [X_{s+1}^T (T_s \beta) + X_{\beta}^T t_{s+1}]$, then normalize w . After each component, deflate X by $X \leftarrow X - t p^T$ with $p = X^T t / (t^T t)$. A joint VAR(s) is fit on the extracted score matrix for prediction utilities.

Value

A `bi_projector` object of class `dicca` with fields:

- `v` - weight matrix ($m \times l$)
- `s` - score matrix ($T \times l$)
- `sdev` - standard deviations of components
- `preproc` - fitted preprocessor
- `loadings` - loading matrix ($m \times l$)
- `betas` - beta coefficients ($l \times s$)
- `theta` - VAR coefficients ($(l*s) \times l$) [classic only]
- `R2` - R^2 values per component
- `lag_order` - lag parameter s
- `obj_history` - convergence history
- `iters_per_component` - iterations per component
- `compact_models` - per-component ARIMA models (compact variants only)
- `compact_info` - compact model metadata (compact variants only)

References

Dong & Qin (2018) Dynamic-Inner Canonical Correlation and Causality Analysis for High Dimensional Time Series Data, IFAC ADCHEM.

Examples

```
set.seed(1)
X <- matrix(rnorm(200), 100, 4)
fit <- dicca(X, s = 1, l = 2, inner = "classic", n_init = 1, max_iter = 100)
fit$R2
```

dicca_predict_scores	<i>One-step-ahead prediction of DLVs (diagonal $G(B)$)</i>
----------------------	---

Description

One-step-ahead prediction of DLVs (diagonal $G(B)$)

Usage

```
dicca_predict_scores(model, scores)
```

Arguments

model	a fitted DiCCA model object with compact inner models
scores	numeric matrix of observed scores to predict from

Value

matrix of predicted scores

dicca_scores	<i>Transform new data to Di(L)V scores using a fitted compact DiCCA model</i>
--------------	---

Description

Transform new data to Di(L)V scores using a fitted compact DiCCA model

Usage

```
dicca_scores(model, Xnew)
```

Arguments

model	a fitted DiCCA model object
Xnew	numeric matrix of new data to transform

Value

matrix of scores

dipca	<i>Dynamic Inner PCA (DiPCA)</i>
-------	----------------------------------

Description

Fit dynamic inner principal components using the fast coordinate-maximization algorithm with optional inner power iterations for the w-update.

Usage

```
dipca(  
  X,  
  S,  
  l,  
  preproc = multivarious::center(),  
  tol = 1e-07,  
  max_iter = 1000,  
  n_init = 4,  
  algorithm = c("I", "II"),  
  inner_power = 50,  
  inner_tol = 1e-08,  
  verbose = 0,  
  seed = NULL  
)
```

```
DiPCA(  
  X,  
  S,  
  l,  
  preproc = multivarious::center(),  
  tol = 1e-07,  
  max_iter = 1000,  
  n_init = 4,  
  algorithm = c("I", "II"),  
  inner_power = 50,  
  inner_tol = 1e-08,  
  verbose = 0,  
  seed = NULL  
)
```

```
dipca_fit(  
  X,  
  S,  
  l,  
  preproc = multivarious::center(),  
  tol = 1e-07,  
  max_iter = 1000,
```

```

n_init = 4,
algorithm = c("I", "II"),
inner_power = 50,
inner_tol = 1e-08,
verbose = 0,
seed = NULL
)

```

Arguments

<code>X</code>	numeric matrix of shape (T, m), time by variables.
<code>s</code>	integer, lag order (≥ 1).
<code>l</code>	integer, number of components to extract.
<code>preproc</code>	a <code>pre_processor</code> object from the multivarious package. Default is <code>center()</code> which centers columns. Use <code>prep(pass())</code> for no preprocessing.
<code>tol</code>	numeric, outer stopping tolerance on the eigen residual $\ d - \lambda w\ _\infty$.
<code>max_iter</code>	integer, maximum outer iterations.
<code>n_init</code>	integer, random restarts per component.
<code>algorithm</code>	"I" (one power step) or "II" (inner power iterations).
<code>inner_power</code>	integer, max inner power iterations (algorithm "II").
<code>inner_tol</code>	numeric, tolerance for inner power iteration.
<code>verbose</code>	integer, verbosity (0=quiet).
<code>seed</code>	optional integer RNG seed for reproducibility.

Details

The implementation follows the DiPCA objective and iteration given in Dong & Qin (2018, Table 2; Eqs 8–18) and the coordinate-maximization interpretation and residual criterion $\|d - \lambda w\|_\infty$ in Shin et al. (2020). The code avoids forming Y_i explicitly; all updates are realized via matvecs with lagged blocks X_{s+1-i} .

Value

A `bi_projector` object of class `dipca` with fields:

- `v` - weight matrix (m x l)
- `s` - score matrix (T x l)
- `sdev` - standard deviations of components
- `preproc` - fitted preprocessor
- `loadings` - loading matrix (m x l)
- `betas` - beta coefficients (l x s)
- `theta` - VAR coefficients ((l*s) x l)
- `lag_order` - lag parameter s
- `obj_history` - convergence history
- `iters_per_component` - iterations per component

Examples

```
set.seed(1)
X <- matrix(rnorm(200), 100, 4)
fit <- dipca(X, s = 1, l = 2, n_init = 1, max_iter = 100)
head(multivarious::scores(fit))
```

dipls

Dynamic-inner Partial Least Squares (DiPLS)

Description

Fit the Dynamic-inner PLS algorithm of Dong & Qin (2018) to jointly model input block X_t and output block Y_t . The outer loop updates the dynamic weights w , q , and FIR coefficients β following Appendix A of the paper, while the inner model is estimated either as a FIR filter (default) or an ARX refinement.

Usage

```
dipls(
  X,
  Y,
  n_comp,
  s,
  preproc_x = multivarious::center(),
  preproc_y = NULL,
  mode = c("fir", "arx"),
  max_iter = 200L,
  tol = 1e-07,
  verbose = 1L
)
```

Arguments

<code>X</code>	numeric matrix with T rows (time) and m columns (inputs).
<code>Y</code>	numeric matrix with T rows and p columns (outputs).
<code>n_comp</code>	integer number of dynamic latent components to extract.
<code>s</code>	non-negative integer lag order for the inner model.
<code>preproc_x</code>	a <code>pre_processor</code> object from the <code>multivarious</code> package for the X block. Default is <code>center()</code> which centers columns. Use <code>prep(pass())</code> for no preprocessing.
<code>preproc_y</code>	a <code>pre_processor</code> object from the <code>multivarious</code> package for the Y block. If <code>NULL</code> (default), uses the same preprocessing as <code>preproc_x</code> .
<code>mode</code>	character, either "fir" for the basic DiPLS inner model or "arx" to augment with AR terms on u .
<code>max_iter</code>	integer, maximum outer iterations per component.

tol	numeric tolerance on the maximum change of w , q , and β between outer iterations.
verbose	integer verbosity level (0 = quiet, 1 = per-component banner, >1 also logs iteration deltas).

Value

A cross_projector object of class dipls with fields:

- vx - X-block weight matrix (m x n_comp)
- vy - Y-block weight matrix (p x n_comp)
- preproc_x - fitted X-block preprocessor
- preproc_y - fitted Y-block preprocessor
- T - X-block score matrix (T x n_comp)
- U - Y-block score matrix (T x n_comp), first s rows are NA
- P - X-block loadings (m x n_comp)
- C - Y-block loadings (p x n_comp)
- inner - per-component inner model coefficients (list)
- R - X-block projection matrix
- lag_order - lag parameter s
- mode - inner model mode ("fir" or "arx")
- iterations - iteration counts per component

Examples

```
set.seed(1)
X <- matrix(rnorm(200), 100, 4)
Y <- matrix(rnorm(200), 100, 2)
fit <- dipls(X, Y, n_comp = 1, s = 1, mode = "fir", verbose = 0)
dim(predict(fit, X))
```

dipls_scores	<i>Project X to DiPLS scores</i>
--------------	----------------------------------

Description

Project X to DiPLS scores

Usage

```
dipls_scores(model, Xnew)
```

Arguments

model	fitted DiPLS object from dipls .
Xnew	numeric matrix with the same number of columns as the training X.

Value

Matrix of latent scores (rows correspond to observations).

predict.dicca	<i>Predict Method for DiCCA</i>
---------------	---------------------------------

Description

Temporal forecasting using the VAR model (classic) or per-component ARIMA models (compact).

Usage

```
## S3 method for class 'dicca'
predict(object, newdata = NULL, ...)
```

Arguments

object	A fitted dicca object (bi_projector with class "dicca")
newdata	New data matrix to predict from
...	Additional arguments (currently unused)

Details

This method projects new data to the latent space and applies the fitted temporal model (VAR for classic, ARIMA for compact variants).

Value

A list with components:

- scores - Observed latent scores
- scores_hat - Predicted latent scores

predict.dipca	<i>Predict Method for DiPCA</i>
---------------	---------------------------------

Description

Temporal forecasting using the VAR model fitted on latent scores.

Usage

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

Arguments

object	A fitted dipca object (bi_projector with class "dipca")
newdata	New data matrix to predict from
...	Additional arguments (currently unused)

Details

This method projects new data to the latent space and applies the fitted VAR(s) model to generate one-step-ahead predictions. The first s time points have NA predictions since they require lagged values.

Value

A list with components:

- scores - Observed latent scores
- scores_hat - Predicted latent scores using VAR(s) model

predict.dipls	<i>Predict Y using a fitted DiPLS model</i>
---------------	---

Description

Predict Y using a fitted DiPLS model

Usage

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

Arguments

object	fitted "dipls" object.
newdata	matrix X or a list containing an element X.
...	unused.

Value

Matrix of one-step-ahead predictions for Y.

residuals.dicca	<i>Residuals Method for DiCCA</i>
-----------------	-----------------------------------

Description

Compute temporal prediction residuals for DiCCA model.

Usage

```
## S3 method for class 'dicca'
residuals(object, newdata = NULL, ...)
```

Arguments

object	A fitted dicca object (bi_projector with class "dicca")
newdata	New data matrix (optional, uses fitted data if missing)
...	Additional arguments (currently unused)

Details

Computes both latent score prediction errors from the VAR/ARIMA model and data-space reconstruction errors (dynamic whitening filter output). The first s time points have NA residuals since predictions require lagged values.

Value

A list with components:

- v - Score-space residuals ($T_t - \hat{T}_t$)
- e_hat - Data-space reconstruction residuals ($X_t - \hat{X}_t$)
- scores - Observed latent scores

residuals.dipca	<i>Residuals Method for DiPCA</i>
-----------------	-----------------------------------

Description

Compute temporal prediction residuals for DiPCA model.

Usage

```
## S3 method for class 'dipca'
residuals(object, newdata = NULL, ...)
```

Arguments

object	A fitted dipca object (bi_projector with class "dipca")
newdata	New data matrix (optional, uses fitted data if missing)
...	Additional arguments (currently unused)

Details

Computes both latent score prediction errors from the VAR model and data-space reconstruction errors. The first s time points have NA residuals since predictions require lagged values.

Value

A list with components:

- v - Score-space residuals ($T_t - \hat{T}_t$)
- e_hat - Data-space reconstruction residuals ($X_t - \hat{X}_t$)
- $scores$ - Observed latent scores

residuals.dipls	<i>Compute residuals for DiPLS model</i>
-----------------	--

Description

Compute residuals for DiPLS model

Usage

```
## S3 method for class 'dipls'
residuals(object, newdata = NULL, ...)
```

Arguments

object	fitted "dipls" object.
newdata	optional list with elements X and Y. If NULL, uses training data (if stored).
...	unused.

Value

Matrix of residuals ($Y - \hat{Y}$).

Index

arima_dicca, [2](#)

DiCCA (dicca), [3](#)
dicca, [3](#)
dicca_predict_scores, [6](#)
dicca_scores, [6](#)
DiPCA (dipca), [7](#)
dipca, [7](#)
dipca_fit (dipca), [7](#)
dipls, [9](#), [11](#)
dipls_scores, [10](#)

predict.dicca, [11](#)
predict.dipca, [12](#)
predict.dipls, [12](#)

residuals.dicca, [13](#)
residuals.dipca, [14](#)
residuals.dipls, [14](#)