Package 'kpcaIG'

July 21, 2024

Title Variables Interpretability with Kernel PCA

Version 1.0

Maintainer Mitja Briscik <mitja.briscik@math.univ-toulouse.fr>

Description

The kernelized version of principal component analysis (KPCA) has proven to be a valid nonlinear alternative for tackling the nonlinearity of biological sample spaces. However, it poses new challenges in terms of the interpretability of the original variables. 'kpcaIG' aims to provide a tool to select the most relevant variables based on the kernel PCA representation of the data as in Briscik et al. (2023) <doi:10.1186/s12859-023-05404-y>. It also includes functions for 2D and 3D visualization of the original variables (as arrows) into the kernel principal components axes, highlighting the contribution of the most important ones.

License GPL-3

Encoding UTF-8

Imports grDevices, rgl, kernlab, ggplot2, stats, progress, viridis, WallomicsData

NeedsCompilation no

Author Mitja Briscik [aut, cre],
Mohamed Heimida [aut],
Sébastien Déjean [aut]

Repository CRAN

Date/Publication 2024-07-21 08:50:05 UTC

Contents

`kernelpca` *Kernel Principal Components Analysis*

Description

Kernel Principal Components Analysis, a nonlinear version of principal component analysis obtrained through the so-called kernel trick.

Usage

```
kernelpca(data, kernel = "vanilladot", kpar = list(), features = 0)
```

Arguments

`data` The data matrix organized by rows. Users should scale the data appropriately before applying this function, if relevant.

`kernel` The kernel function used for the analysis. It can be chosen from the following strings:

- `'rbfdot'`: Radial Basis kernel function "Gaussian"
- `'polydot'`: Polynomial kernel function
- `'vanilladot'`: Linear kernel function
- `'tanhdot'`: Hyperbolic tangent kernel function

`kpar` The list of hyper-parameters (kernel parameters) used with the kernel function. The valid parameters for each kernel type are as follows:

- `sigma`: inverse kernel width for the Radial Basis kernel function `"rbfdot"`.
- `degree`, `scale`, `offset` for the Polynomial kernel function `"polydot"`.
- `scale`, `offset` for the Hyperbolic tangent kernel function `"tanhdot"`.

`features` The number of features (kernel principal components) to use for the analysis. Default: 0 , (all)

Value

`kernelpca` returns an S4 object of formal class kpca as in library(kernlab) containing the principal component vectors along with the corresponding eigenvalues.

`pcv` pcv a matrix containing the principal component vectors (column wise)

`eig` The corresponding eigenvalues

`rotated` The original data projected (rotated) on the principal components

`xmatrix` The original data matrix

References

Scholkopf B., Smola A. and Muller K.R. (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, 10, 1299-1319.

Examples

```
# Example
library(WallomicsData)
library(kpcaIG)
library(ggplot2)
library(kernlab)
Transcriptomics_Stems_s <- scale(Transcriptomics_Stems)

kpca_tan <-  kernelpca(as.matrix(Transcriptomics_Stems_s),
                          kernel = "tanhdot",
                          kpar = list(scale = 0.0001, offset = 0.01))


ggplot(data = data.frame(rotated(kpca_tan), Genetic_Cluster),
       aes(x = X1, y = X2, shape = Genetic_Cluster)) +
  geom_point(size = 2, aes(color = Genetic_Cluster)) +
  xlab("1st kernel PC") +
  ylab("2nd kernel PC") +
  labs(color = "Genetic_Cluster", shape = "Genetic_Cluster") +
  theme_minimal()


ggplot(data = data.frame(rotated(kpca_tan), Ecotype),
       aes(x = X1, y = X2, shape =  Ecotype)) +
  geom_point(size = 2, aes(color =  Ecotype)) +
  xlab("1st kernel PC") +
  ylab("2nd kernel PC") +
  labs(color = " Ecotype", shape = " Ecotype") +
  theme_minimal()
```

`kpca_igrad` *KPCA-IG: Variables Interpretability in Kernel PCA*

Description

KPCA-IG, kernel pca interpetable gradient. It is the fuction that gives the feature ranking, from the most to the least relevant variable. The ranking is obtained through the kernel's partial derivatives computation. A score, which corresponds to the score mean among the sample points, is assigned to each input feature.

Usage

```
kpca_igrad(kpca_result, dim, mean_type = "arithmetic", trim_ratio = 0.1)
```

Arguments

`kpca_result` The result of the previously obtained kernel PCA analysis.

`dim` Number of kernel principal component to use for the computation of the scores. It should be less or equal to the number of component of the kPCA.

`mean_type` Type of mean. Possible values are "arithmetic", "geometric", "harmonic", "median", or "trimmed". Default = "arithmetic"

`trim_ratio` For mean_type == "trimmed", it is the fraction (0 to 0.5) of scores to be trimmed from each end before the mean is computed for a more robust to outliers arithmetic mean computation.

Value

A data frame containing the sorted variables and their scores sorted in decreasing order.

References

Briscik, M., Dillies, MA. & Déjean, S. Improvement of variables interpretability in kernel PCA. BMC Bioinformatics 24, 282 (2023). DOI: doi:10.1186/s1285902305404y

Examples

```
library(WallomicsData)
library(kpcaIG)

Transcriptomics_Stems_s <- scale(Transcriptomics_Stems)

kpca_tan <-  kernelpca(as.matrix(Transcriptomics_Stems_s),
                          kernel = "tanhdot",
                          kpar = list(scale = 0.0001, offset = 0.01))

#Compute the most relevant genes based on the first two components of kpca_tan

kpca_ig_tan <- kpca_igrad(kpca_tan, dim = c(1,2))
head(kpca_ig_tan)
```

`plot_kpca2D` *2D Kernel PCA Plot with Variables Representation*

Description

`plot_kpca2D` allows to visualize an original variable of interest in the specified principal components. The variable is displayed as an arrow, showing its relevance in the relative position of each sample point in the kernel component space.

Usage

```
plot_kpca2D(kpca_result, target_variable, groups = NULL, scale = 100,
  components = c(1, 2), arrow_col = "#D3D3D3",
  main_title = "Kernel principal component analysis")
```

Arguments

`kpca_result` The result of the previously obtained kernel PCA analysis

`target_variable`
A string indicating the name of the variable of interest to visualize as arrows on the kernel PCA plot.

`groups` A vector indicating the grouping of data points, if applicable. Default: NULL

`scale` Coefficient to adjust the lengths of the arrows. Default: 100

`components` A numeric vector of length 2 specifying the indices of the components to plot. Default: c(1, 2)

`arrow_col` Colour of the arrows. Default: '#D3D3D3'

`main_title` Graph title. Default: "Kernel principal component analysis"

Value

Provides a 2D plot of class ggplot that displays the sample points projected onto the specified kernel principal component axes, with the variables of interest represented as arrows.

References

Briscik, M., Dillies, MA. & Déjean, S. Improvement of variables interpretability in kernel PCA. BMC Bioinformatics 24, 282 (2023). DOI: doi:10.1186/s1285902305404y. Variables representation as in Reverter, F., Vegas, E. & Oller, J.M. Kernel-PCA data integration with enhanced interpretability. BMC Syst Biol 8 (Suppl 2), S6 (2014). DOI: `doi:10.1186/1752-0509-8-S2-S6`

Examples

```
library(WallomicsData)
library(kpcaIG)

Transcriptomics_Stems_s <- scale(Transcriptomics_Stems)

kpca_tan <-  kernelpca(as.matrix(Transcriptomics_Stems_s),
                        kernel = "tanhdot",
                        kpar = list(scale = 0.0001, offset = 0.01))

# Compute the most relevant genes based on the first two components of kpca_tan

kpca_ig_tan <- kpca_igrad(kpca_tan, dim = c(1,2))
head(kpca_ig_tan)

# Visualize the most relevant variable (gene) according to kpca_igrad, "AT4G12060".
```

```
plot_kpca2D(kpca_tan, "AT4G12060", groups = Ecotype, scale = 1000, components = c(1, 2))

# Visualize using the second and third components

plot_kpca2D(kpca_tan, "AT4G12060", groups = Ecotype, scale = 1000, components = c(2, 3))

#The selected gene shows upper expression in the samples with genotype type Col.
```

plot_kpca3D *3D Kernel PCA Plot with Variables Representation*

Description

plot_kpca3D allows to visualize an original variable of interest in the first three principal components. The variable is displayed as an arrow, showing its relevance in the relative position of each sample point in the kernel component space.

Usage

```
plot_kpca3D(kpca_result, target_variable, groups, scale=1,
type = "s", size = 3/4, arrow_col = "#999999",
angles = 12, main = NULL)
```

Arguments

kpca_result The result of the previously obtained kernel PCA analysis.

target_variable A string indicating the name of the variable to visualize as arrows on the kernel PCA plot.

groups A vector indicating the grouping of data points, if applicable. Default: NULL

scale Coefficient to adjust the lengths of the arrows. Default 1

type A character indicating the type of point for the observations. Supported types are: ’p’ for points, ’s’ for spheres. Default: ’s’

size The size of the plotted points. Default: 3/4

arrow_col Colour of the arrows. Default: ’#999999

angles Number of barbs of the arrows. Default: 12

main Graph title. Default: NULL

Value

Provides an interactive 3D plot that displays the sample points projected onto the first three kernel principal component axes, with the variables of interest represented as arrows.

References

Briscik, M., Dillies, MA. & Déjean, S. Improvement of variables interpretability in kernel PCA. BMC Bioinformatics 24, 282 (2023). DOI: doi:10.1186/s1285902305404y. Variables representation as in Reverter, F., Vegas, E. & Oller, J.M. Kernel-PCA data integration with enhanced interpretability. BMC Syst Biol 8 (Suppl 2), S6 (2014). DOI: doi:10.1186/1752-0509-8-S2-S6

Examples

```
library(WallomicsData)
library(kpcaIG)

Transcriptomics_Stems_s <- scale(Transcriptomics_Stems)

kpca_tan <-  kernelpca(as.matrix(Transcriptomics_Stems_s),
                          kernel = "tanhdot",
                          kpar = list(scale = 0.0001, offset = 0.01))

#Compute the most relevant genes based on the first two components of kpca_tan

kpca_ig_tan <- kpca_igrad(kpca_tan, dim = c(1,2))
head(kpca_ig_tan)

#Visualize the most relevant variable (gene) according to kpca_igrad, "AT4G12060".

plot_kpca3D(kpca_tan, "AT4G12060", groups = Ecotype, scale = 1000)

#The selected gene shows upper expression in the samples with genotype type Col.
```

Index