Invariant coordinate selection - a guided tour of implementations

Aurore Archimbaud, Colombe Becquart,
Andreas Alfons & Klaus Nordhausen
joint w/ Anne M. Ruiz

ICS and Related Methods Conference
May 8, 2025

Outline of the presentation

Invariant coordinate selection
- Principle
- Interpretation of the components
- Summary

ICS - methodology

ICS - implementations

Conclusion

Tutorial

Invariant Component Selection (ICS)

ICS Tyler et al. (2009) - GPCA Caussinus and Ruiz-Gazen (1990)

Main characteristics

- Simultaneously diagonalizes two scatter matrices.
- Affine invariant.
- Dimension reduction method useful for multivariate data analysis.

Invariant Component Selection (ICS)

Main advantages

- Goes beyond PCA, which diagonalizes the covariance matrix.
- Can recover the Fisher's linear discriminant subspace under some elliptical mixture models, without knowing the class labels (Tyler et al., 2009; Becquart et al., 2024).

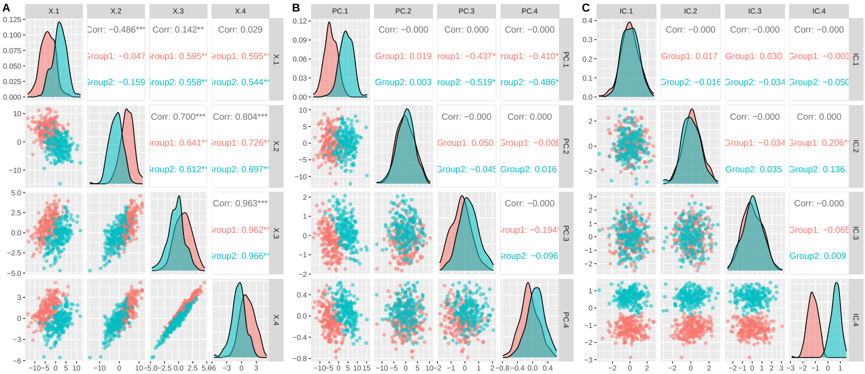
Invariant Component Selection (ICS)

Main advantages

- Goes beyond PCA, which diagonalizes the covariance matrix.
- Can recover the Fisher's linear discriminant subspace under some elliptical mixture models, without knowing the class labels (Tyler et al., 2009; Becquart et al., 2024).

Main steps

1. Components calculation
2. Components selection

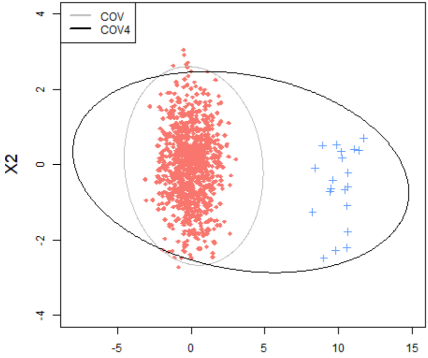


See the toy example from Nordhausen and Ruiz-Gazen (2021).

MEAN-SHIFT MODEL ($q = 1$)

$$980 \text{ obs } \sim \mathcal{N}_{p=2}(0, \mathbf{I}_p) \;\&\; 20 \text{ obs } \sim \mathcal{N}_{p=2}((10,0)', \mathbf{I}_p)$$

Suppose $\mathbf{V}_1 = \mathrm{COV}(\mathbf{X})$ and $\mathbf{V}_2 = \mathrm{COV}_4(\mathbf{X})$
with $\mathrm{COV}_4 = \frac{1}{p+2}\,\mathbb{E}\left[r^2(\mathbf{X})(\mathbf{X} - \mathbb{E}(\mathbf{X}))(\mathbf{X} - \mathbb{E}(\mathbf{X}))'\right]$.



Example: ICS with COV-COV4

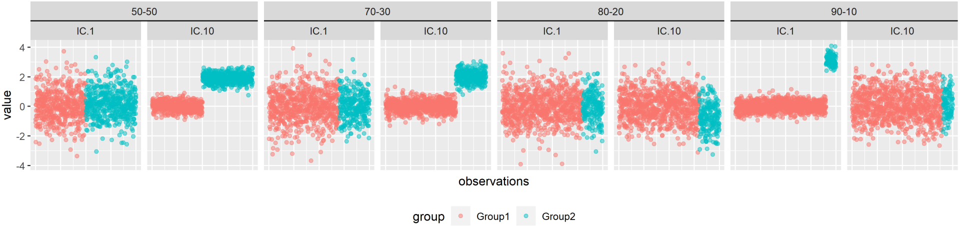


Figure: Mixture of 2 groups: first and last ICs

1. Components calculation

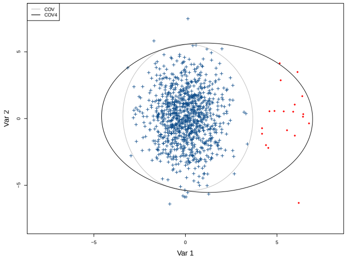
1. Two scatter matrices $\mathbf{V}_1$ and $\mathbf{V}_2$.
2. Eigendecomposition of $\mathbf{V}_1^{-1}\mathbf{V}_2$.
3. Decreasing eigenvalues: kurtosis.
4. Projection of the centered data on the eigenvectors $\rightarrow$ ICs.

1. Components calculation

1. $\mathbf{V}_1$: covariance matrix.
2. $\mathbf{V}_2$: fourth moment matrix.
3. Eigendecomposition of $\mathbf{V}_1^{-1}\mathbf{V}_2$
4. Projection of the centered data on the eigenvectors.

1. Components calculation

1. $\mathbf{V}_1$: covariance matrix.
2. $\mathbf{V}_2$: fourth moment matrix.
3. Eigendecomposition of $\mathbf{V}_1^{-1}\mathbf{V}_2$
4. Projection of the centered data on the eigenvectors.

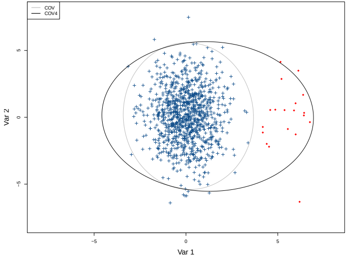


1. Components calculation

1. $\mathbf{V}_1$: covariance matrix.
2. $\mathbf{V}_2$: fourth moment matrix.
3. Eigendecomposition of $\mathbf{V}_1^{-1}\mathbf{V}_2$
4. Projection of the centered data on the eigenvectors.

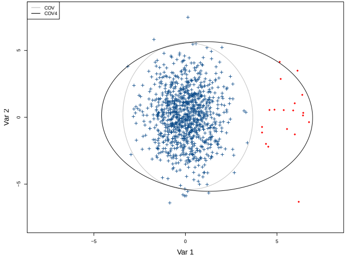
2. Components selection

1. Based on the eigenvalues.
2. Based on the components.
3. First and/or last $k \leq p$ components retained.



1. Components calculation

1. $\mathbf{V}_1$: covariance matrix.
2. $\mathbf{V}_2$: fourth moment matrix.
3. Eigendecomposition of $\mathbf{V}_1^{-1}\mathbf{V}_2$
4. Projection of the centered data on the eigenvectors.



2. Components selection

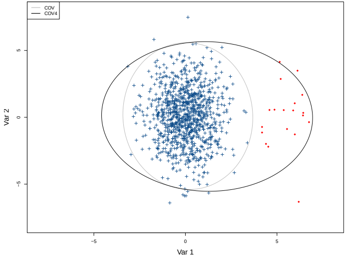
1. Based on the eigenvalues.
2. Based on the components.
3. First and/or last $k \leq p$ components retained.

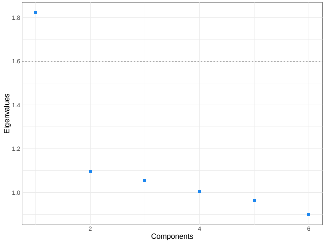
1. Components calculation

1. $\mathbf{V}_1$: covariance matrix.
2. $\mathbf{V}_2$: fourth moment matrix.
3. Eigendecomposition of $\mathbf{V}_1^{-1}\mathbf{V}_2$
4. Projection of the centered data on the eigenvectors.

2. Components selection

1. Based on the eigenvalues.
2. Based on the components.
3. First and/or last $k \leq p$ components retained.





Applications

Outlier detection

1. Euclidian distance of the observations calculated using the k selected components.

Applications

Outlier detection

1. Euclidian distance of the observations calculated using the k selected components.

   Remark: $k = p \rightarrow$ Mahalanobis distance w.r.t. $\mathbf{V}_1$

Applications

Outlier detection

1. Euclidian distance of the observations calculated using the k selected components.

   Remark: $k = p \rightarrow$ Mahalanobis distance w.r.t. $\mathbf{V}_1$
2. Cutoff based on quantiles from simulations.

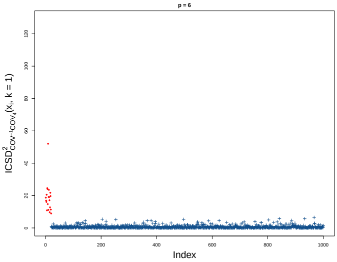
Applications

Outlier detection

1. Euclidian distance of the observations calculated using the k selected components.

   Remark: $k = p \rightarrow$ Mahalanobis distance w.r.t. $\mathbf{V}_1$

2. Cutoff based on quantiles from simulations.



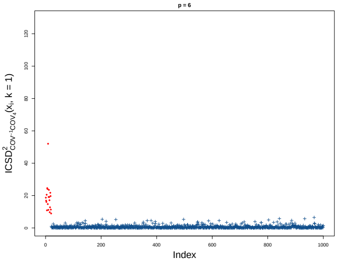
Applications

Outlier detection

1. Euclidian distance of the observations calculated using the k selected components.

   Remark: $k = p \rightarrow$ Mahalanobis distance w.r.t. $\mathbf{V}_1$

2. Cutoff based on quantiles from simulations.



Tandem clustering with ICS

1. Clustering method.
2. Identification of clusters.

Applications

Outlier detection

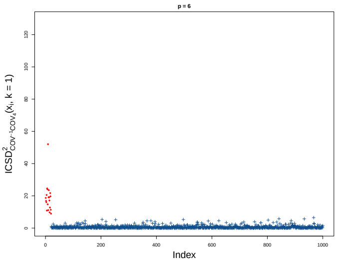
1. Euclidian distance of the observations calculated using the k selected components.

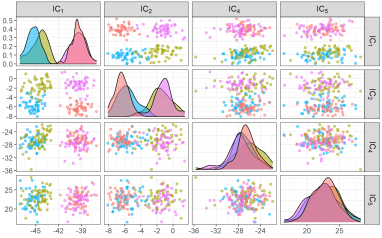
   Remark: $k = p \rightarrow$ Mahalanobis distance w.r.t. $\mathbf{V}_1$

2. Cutoff based on quantiles from simulations.

Tandem clustering with ICS

1. Clustering method.
2. Identification of clusters.





Outline of the presentation

Invariant coordinate selection

ICS - methodology

Choice of scatter matrices

Choice of components

ICS - implementations

Conclusion

Tutorial

Choice of scatter matrices

Applications:

- For **outlier detection**: Archimbaud et al. (2018a) for a small proportion of outliers.

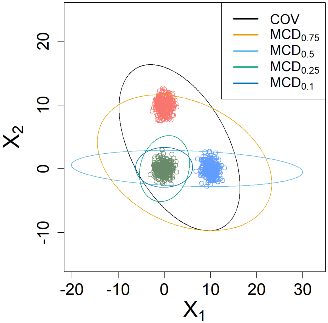
Choice of scatter matrices

Applications:

- For **outlier detection**: Archimbaud et al. (2018a) for a small proportion of outliers.
- For **clustering**: Becquart et al. (2024) and Alfons et al. (2024). One scatter matrix should reflect the within (local) cluster structure and one the between (global) cluster structure.

Choice of scatter matrices

- COV – COV_4 or FOBI Peña et al. (2010); Cardoso (1989); Nordhausen and Virta (2019); Fischer et al. (2017, 2020)
- MCD_α – COV, RMCD_α – COV with values of $\alpha \leq 0.5$ Rousseeuw (1985)
- TCOV – UCOV, TCOV – COV Caussinus and Ruiz-Gazen (1993a)
- LCOV – COV Hennig (2009)



Description	R - Scatter	R - ICS_scatter	Python `Scatter`
Covariance	`ICS::cov()`	`ICS::ICS_cov()`	`cov()`
Fourth-moment covariance	`ICS::cov4()`	`ICS::ICS_cov4()`	`cov4()`
One-step M-estimator	`ICS:: covW()`	`ICS::ICS_covW()`	`covW()`
One-step Tyler shape matrix	`ICS::covAxis()`	`ICS::ICS_covAxis()`	`covAxis()`
Multivariate t-distribution estimator	`ICS:: tM()`	`ICS:: ICS_tM()`	
Supervised scatter (quantiles)	`ICS::scovq()`	`ICS::ICS_scovq()`	
Minimum Covariance Determinant (MCD)	`rrcov::CovMcd()`	`ICSClust::ICS_mcd_raw()`, `ICS_mcd_rwt()`	`sklearn.covariance.MinCovDet()`
Cauchy location and scatter	`ICS::tM()`	`ICSClust::ICS_mlc()`	
Pairwise one-step M-estimate	`ICSClust::tcov()`	`ICSClust::ICS_tcov()`	
Simple robust estimates	`ICSClust::ucov()`	`ICSClust::ICS_ucov()`	
Local shape scatter	`ICSClust::lcov()`	`ICSClust::ICS_lcov()`	

CHOICE OF COMPONENTS WITH $q < p$

For outlier detection:

- First components

CHOICE OF COMPONENTS WITH $q < p$

For outlier detection:

- First components

For clustering:

- First and/or last components

CHOICE OF COMPONENTS WITH $q < p$

Archimbaud et al. (2018a), Alfons et al. (2024), Nordhausen et al. (2022), Radojicic and Nordhausen (2020)

Choice of components with $q < p$

Archimbaud et al. (2018a), Alfons et al. (2024), Nordhausen et al. (2022), Radojicic and Nordhausen (2020)

Based on the Invariant Components:

- Keep only non-gaussian components using marginal normality tests.

Choice of components with $q < p$

Archimbaud et al. (2018a), Alfons et al. (2024), Nordhausen et al. (2022), Radojicic and Nordhausen (2020)

Based on the Invariant Components:

- Keep only non-gaussian components using marginal normality tests.

In this context of particular sequential multiple testing, we apply the Bonferroni correction on the significance level: $\alpha_i = \alpha / i$ for $i = 1, \ldots, p$ with $\alpha = 5\%$ (see Dray (2008)).

Choice of components with $q < p$

Based on the eigenvalues and eigenvectors:

- Visually, using a scree plot.
- Using the number q of clusters a priori and the fact that some should be almost equal (med_criterion and var_criterion).
- Using asymptotic distribution of the eigenvalues or bootstrapping (ICSboot, FOBIasymp, FOBIboot).
- Using quasi inferential procedures (parallel analysis).
- Using ladle and data augmentation (FOBIladle).

Selection of components - summary

Criterion	Function	Notes
Marginal normality tests	`ICSClust::normal_crit()`	First and last few components are investigated
Median-based criterion	`ICSClust::med_crit()`	Requires a priori knowledge
Variance-based criterion	`ICSClust::var_crit()`	From `ICtest::ICSboot()`
Discriminatory power criterion	`ICSClust::discriminatory_crit()`	Supervised, requires a priori knowledge
Parallel analysis (simulation)	`ICSOutlier::comp_simu_test()`, `comp.simu.test()`	Only first components are investigated
Marginal normality tests	`ICSOutlier::comp_norm_test()`, `comp.norm.test()`	Only first components are investigated
Asymptotic and resampling based approaches	`ICtest:: ICSboot()`, `FOBIboot()`, `FOBIasymp()`, `FOBIladle()`	Focus on identifying Gaussian subspace

Outline of the presentation

Invariant coordinate selection

ICS - methodology

ICS - implementations
- R ecosystem
- Python ecosystem
- Julia ecosystem

Conclusion

Tutorial

Genesis: 2007/2008 - Klaus Nordhausen + Hannu Oja + David E. Tyler

ICS (v1.0-1.2)

ics (S4)

```
ics(X, S1 = cov, S2 = cov4, S1args = list(), S2args = list(),
    stdB = "Z", stdKurt = TRUE, na.action = na.fail)
```

```
@gKurt, @UnMix, @Scores
```

Scatter matrices:

```
cov4, cov4.wt,
covAxis, covOrigin, tM
```

2016-2018: + Aurore Archimbaud and Anne Ruiz-Gazen

+ Joris May

ICSShiny (v0.4-0.5)

ICSShiny (GUI)

1. *Choice of the scatter matrices*
2. *Selection of the ICs*
3. *Exploration of the output and identification of outliers*
4. *Download of graphs and components.*

ICSOutlier (v2.0-3.0)

ics.outlier (S4)

1. *Selection of ICs:* `comp.simu.test` `comp.norm.test`
2. *ICS distances:* `ics.distances`
3. *Identification of outliers:* `dist.simu.test`

ICtest (v2.0-3-1)

1. *Selection of ICs*

ICS (v1.3)

ics (S4)

```
ics(X, S1 = cov, S2 = cov4, S1args = list(), S2args = list(),
    stdB = "Z", stdKurt = TRUE, na.action = na.fail)
```

`@gKurt, @UnMix, @Scores`

ics2 (S4)

```
ics2(X, S1 = MeanCov, S2 = Mean3Cov4, S1args = list(), S2args = list(),
     na.action = na.fail)
```

two location functionals used to fix the signs of the components and to center them.

2023: + Andreas Alfons + Zlatko Drmač

ICSClust (v1.0)

ICSClust(S3)

ICS_scatter : `ICS_lcov, ICS_mcd, ICS_mlc, ICS_tcov, ICS_ucov`
1. Computation of ICS: `ICS`
2. Selection of ICs and visualization:
`"med_crit", "normal_crit", "var_crit", "discriminatory_crit"`
3. Clustering and visualization :
`"kmeans_clust", "tkmeans_clust", "pam_clust", "mclust_clust", "rmclust_clust", "rimle_clust"`

ICSOutlier (v4.0)

ICS_outlier(S3)

0. Computation of ICS: `ICS`
1. Selection of ICs: `comp_simu_test comp_norm_test`
2. ICS distances: `ics_distances`
3. Identification of outliers: `dist_simu_test`

ICS (v1.4-2)

```
ICS(
  X,
  S1 = ICS_cov,
  S2 = ICS_cov4,
  S1_args = list(),
  S2_args = list(),
  algorithm = c("whiten", "standard", "QR"),
  center = FALSE,
  fix_signs = c("scores", "W"),
  na.action = na.fail
)
```

```
@gKurt   → $gen_kurtosis
@UnMix   → $W
@Scores  → $scores
```

ICS (S3)

1. `ICS_scatter` *class*
2. *Different algorithms*
3. *Centering*
4. *Fixing signs*

ICS - ALGORITHMS

Let $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ be a p-variate dataset.
Simultaneous diagonalization of two scatter matrices $\mathbf{S}_{1,n}$ and $\mathbf{S}_{2,n}$:

$$\mathbf{W}_n \mathbf{S}_{1,n} \mathbf{W}_n^\top = \mathbf{I}_p \quad \text{and} \quad \mathbf{W}_n \mathbf{S}_{2,n} \mathbf{W}_n^\top = \mathbf{D}_n$$

where the diagonal matrix $\mathbf{D}_n$ contains the eigenvalues $\rho_1, \ldots, \rho_p$ of $\mathbf{S}_{1,n}^{-1}\mathbf{S}_{2,n}$ in decreasing order and $\mathbf{W}_n = (\mathbf{w}_1, \ldots, \mathbf{w}_p)'$ contains the corresponding eigenvectors as its rows.

ICS - ALGORITHMS

Let $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ be a p-variate dataset.
Simultaneous diagonalization of two scatter matrices $\mathbf{S}_{1,n}$ and $\mathbf{S}_{2,n}$:

$$\mathbf{W}_n \mathbf{S}_{1,n} \mathbf{W}_n^\top = \mathbf{I}_p \quad \text{and} \quad \mathbf{W}_n \mathbf{S}_{2,n} \mathbf{W}_n^\top = \mathbf{D}_n$$

where the diagonal matrix $\mathbf{D}_n$ contains the eigenvalues $\rho_1, \ldots, \rho_p$ of $\mathbf{S}_{1,n}^{-1}\mathbf{S}_{2,n}$ in decreasing order and $\mathbf{W}_n = (\mathbf{w}_1, \ldots, \mathbf{w}_p)'$ contains the corresponding eigenvectors as its rows.
The (affine) invariant coordinates or scores are:

$$\mathbf{Z}_n = \mathbf{X}_n^c \mathbf{W}_n^\top.$$

$\mathbf{X}_n^c$ can be the centered version of $\mathbf{X}_n$ with respect to the location estimator associated with $\mathbf{S}_{1,n}$.

ICS - ALGORITHMS

- **whiten**: whitens X_n with respect to $\mathbf{S}_1$ before computing $\mathbf{S}_2$ (should be a function).
- **standard**: performs the spectral decomposition of the symmetric matrix $M(X_n)$.

ICS - ALGORITHMS

- **whiten**: whitens X_n with respect to $\mathbf{S}_1$ before computing $\mathbf{S}_2$ (should be a function).
- **standard**: performs the spectral decomposition of the symmetric matrix $M(X_n)$.

whiten:

- $Y_n = X_n S_1(X_n)^{-1/2}$
- $S_2(Y_n)$
- $S_2(Y_n) = UDU'$
- $W = U' S_1(X_n)^{-1/2}$
- $Z = X_n^c W^\top$

standard:

- $S_1(X_n), S_2(X_n)$
- $M(X_n) = S_1(X_n)^{-1/2} S_2(X_n) S_1(X_n)^{-1/2}$
- $M(X_n) = UDU'$
- $W = U' S_1(X_n)^{-1/2}$
- $Z = X_n^c W^\top$

ICS - ALGORITHMS

The signs of W can be fixed with `fix_signs`:

- **`"scores"`**: the generalized skewness values of all components are positive.
  Common for ICS framework.
- **`"W"`**: the maximum element in each row of W is positive and each row has norm 1.
  Common with ICA framework.

Z can be centered through the boolean `center` input parameter.

ICS - QR - based on pivoted QR decomposition

Archimbaud et al. (2023b), numerically stable algorithm for $\text{COV} - \text{COV}_w$ focusing on:

$$M(X_n) = \text{COV}^{-1/2}\text{COV}_w\text{COV}^{-1/2}.$$

ICS - QR - BASED ON PIVOTED QR DECOMPOSITION

Archimbaud et al. (2023b), numerically stable algorithm for $\mathrm{COV} - \mathrm{COV}_w$ focusing on:

$$M(X_n) = \mathrm{COV}^{-1/2}\mathrm{COV}_w\mathrm{COV}^{-1/2}.$$

Algorithm:

- Pivoted QR factorization: $\Pi_2^\top \left(\frac{1}{\sqrt{n-1}} X_n^{c\top}\right) \Pi_1 = QR$
- $Q = \Pi_2 Q$, $R = R\Pi_1^\top$
- Leverage scores $q_i = \|Q(i,:)\|_2^2$, $i = 1, \ldots, n$
- SVD of Diag $\left(\sqrt{w((n-1)q_i)}\right)_{i=1}^{n} Q$ to obtain

$$\widetilde{M}(X_n) = \frac{n-1}{n} Q^\top \ \mathrm{Diag}(w((n-1)q_i))_{i=1}^{n} \ Q = \widetilde{U}_2\widetilde{D}_2\widetilde{U}_2^\top$$

- $W = (R^{-1}\widetilde{U}_2)^\top$
- $Z = \sqrt{n-1}Q\widetilde{U}_2$ *or equivalently* $Z = X_n^c \Pi_1 W^\top$.

ICS - SUMMARY

Main attributes

Description	R - `ics()`, `ics2()`	R - `ICS()`	Python `ICS`
Generalized eigenvalues	`@gKurt`	`$gen_kurtosis`, `gen_kurtosis()`	`.kurtosis_`
Unmixing matrix.	`@UnMix`	`$W`	`.W_`
Scores	`@Scores`	`$scores`, `components()`	`.scores_`

Methods

Description	R - `ics()`, `ics2()`, `ICS()`	Python `ICS`
Print basic information	`print()`	
Coefficient matrix of ICS	`coef()`	
Summary	`summary()`	`.describe()`
Component scatterplot matrix	`plot()`	`.plot()`
Plots the kurtosis measures	`screeplot()`	`.plot_kurtosis()`

Python - 2023: Colombe Becquart + Abdallah Abdelsameia

Package ICSpyLab

- Created to make ICS accessible to the Python community.
- Main features:
  - the ICS class;
  - several scatter matrices;
  - supported algorithms: standard, whiten, QR.

Python - 2023: Colombe Becquart + Abdallah Abdelsameia

- Main difference from R: computing the invariant coordinates is divided into `fit` and `transform` methods, in line with popular machine learning frameworks such as scikit-learn:
  - `fit` computes the matrix W containing the eigenvectors;
  - `transform` computes the (affine) invariant coordinates or scores with the fitted matrix W. This transformation can be applied to a different dataset than the one used to compute W;
  - use the method `fit_transform` to perform both steps in a single call (equivalent of the function `ICS-S3()` from the R package `ICS`).

Python - 2023: Colombe Becquart + Abdallah Abdelsameia

- Next steps:
  - additional scatter matrices (TCOV);
  - component selection.
- For more details about the implementation, its installation and usage, see the documentation.

Julia - 2023: + Christopher Claassen (EUR)

Master dissertation

- *Generalising Invariant Coordinate Selection to a non-linear dimensionality reduction method* $\Longrightarrow$ work in progress Claassen (2023).
- Julia - code: `https://github.com/CClaassen/SimultaneousDiagonalisation.jl/`.

Julia - 2023: + Christopher Claassen (EUR)

Master dissertation

- *Generalising Invariant Coordinate Selection to a non-linear dimensionality reduction method* $\Longrightarrow$ work in progress Claassen (2023).
- Julia - code: `https://github.com/CClaassen/SimultaneousDiagonalisation.jl/`.

Short description of all source files:

- *SimultaneousDiagonalisation*.*jl* contains the main module of the package.
- *factorisatons*.*jl* contains the main methods to compute PCA and ICS.
- *component_selection*.*jl* contains functions related to component selection.
- *normality_tests*.*jl* contains functions for univariate normality tests.
- *scatters*.*jl* contains functions for computing various scatter matrices.

JULIA - 2023: + CHRISTOPHER CLAASSEN (EUR)

- *smoothing_kernels.jl* contains functions for using smoothing kernels.
- *reproducing_kernels.jl* contains functions for using reproducing kernels.
- *kernel_manipulation.jl* contains functions for transforming kernels.
- *classification.jl* contains functions related to classification.
- *evaluation.jl* contains functions for evaluating classification results.
- *external_methods.jl* contains bindings to external methods.
- *figures.jl* contains functions for plotting different types of figures.
- *data_manipulation.jl* contains functions for transforming data.
- *utilities.jl* contains various useful auxiliary functions.
- *external_data.jl* contains functions for loading external data.
- *experiments.jl* contains functions for replicating all results from the paper.

factorisatons.jl	component_selection.jl	normality_tests.jl	scatters.jl	smoothing_kernels.jl	reproducing_kernels.jl	kernel_manipulation.jl
#Main Methods#	#Main Method#	#Main Method#	#Locations and Scatters#	#Main Methods#	#Main Method#	#Input Transformations#
ics()	**component_selection()**	**normal_test()**	**mean1()**	**kernel_smoother()**	kernel()	scale_input()
gpca()			**cov2()**	**adaptive_kernel_smoother()**		scale_output()
	#Index Methods#	#Moment Tests#	**mean3()**		#Polynomial Kernels#	ard_transform()
#Simultaneous Diagonalisations#	retain_all()	jarque_bera()	**cov4()**	#Limited Support Kernels#	**linear_kernel()**	linear_transform()
REG_EIG()	retain_first()	bonnett_seier()		uniform_kernel_w()	**polynomial_kernel()**	
SYM_EIG()	retain_last()	agostino_pearson()	#Robust Scatters#	triangular_kernel_w()		#Kernel Manipulations#
REG_SVD()		ansecombe_glynn()	**fastMCD()**	epanechnikov_kernel_w()	#Exponential Kernels#	kernel_centered()
SYM_SVD()	#Eigenvalue Methods#	omnibus_K2()	**FastMVE()**	quartic_kernel_w()	abelian_kernel()	kernel_normalized()
	retain_var()			triweight_kernel_w()	**laplacian_kernel()**	kernel2distance()
#Regular Diagonalisations#	kramer_rule()	#ECDF Tests#	#Local Scatters#	tricube_kernel_w()	**gaussian_kernel()**	
EIG()	cluster_priori()	anderson_darling()	**lcov()**	cosine_kernel_w()	gibbs_kernel()	#Nystrom Appoximations#
SVD()		kolmogorov_smirnov()	**rlcov()**		gamma_exponential_kernel()	nystrom_ind()
	#Component Methods#	lilliefors()	**alcov()**	#Infinite Support Kernels#	exponentiated_kernel()	nystrom_ratio()
#Alternative Diagonalisations#	normality()	cramer_mises()	**arlcov()**	gaussian_kernel_w()		
custom_eig()	pick_tsne()	watson()		logistic_kernel_w()	#Rational kernels#	#Kernel Combinations#
custom_svd()	marginal_div()		#Auxiliary Functions#	sigmoid_kernel_w()	**rational_kernel()**	kernel_sum()
reduced_svd()	joint_div()	#Misc Tests#	*opt_h()*	silverman_kernel_w()	rational_quadratic_kernel()	kernel_prod()
custom_gsvd()	batch_joint_div()	shapiro_wilk()	*breakdown()*		gamma_rational_kernel()	
		shapiro_francia()		#Aliases#		#Scale Parameter Settings#
#Rank Reducing Methods#	#t-SNE Loss Functions#	pearson_chi2()		parabolic_kernel_w()	#Periodic Kernels#	**median_trick()**
constrained_svd()	*tsne_loss()*			biweight_kernel_w()	**cosine_kernel()**	**quantile_trick()**
reduce_mat_svd()	*opt_beta()*	#Data Transformations#			**neural_network_kernel()**	
reduce_mat_qr()	*Hbeta()*	*z_score()*			periodic_kernel()	
		rob_score()				
	#Divergences#	*mad_n()*			#Misc Kernels#	
	kl_div()				fbm_kernel()	
	sym_kl_div()	#Auxiliary Functions#			gabor_kernel()	
	gen_kl_div()	*skewness_moments()*			matern_kernel()	
	renyi_div()	*kurtosis_moments()*			wiener_kernel()	
	js_div()	*round_retain_sum()*			**mahalanobis_kernel()**	
		count_sample_regions()				
		geary_kurt()				

classification.jl	evaluation.jl	external_methods.jl	figures.jl	data_manipulation.jl	utilities.jl	external_data.jl	experiments.jl
#Main Methods#	#Cross-validation#	#Visualisation Methods#	#Pairwise Scatters#	#Moment Calculations#	#Diagonal Matrix Shorthands#	#Data Retrieval#	#Thesis Reproduction#
ols2()	**k_fold()**	**tsne2()**	**scatter_plot()**	*raw_moment()*	*eye()*	iris_data()	all_experiments()
logit2()	**stratified_k_fold()**	**umap2()**	**scatter_plot_ind()**	*central_moment()*	*diag_eye()*	word2vec_data()	iris_experiments()
svm2()				*standard_moment()*	*diag_eye_nan()*	glove_data()	wine_experiments()
	#Auxiliary Functions#	#Robust Scatters from R#	#Pairwise Contour Scatters#			fasttext_data()	wbc_experiments()
#Used for Logit#	*split_data_ind()*	FastMCD2()	**contour_plot()**	#Weighted Locations#	#Matrix Shorthands#	ODDS_data()	word2vec_experiments()
loglike()	*stratified_split_data_ind()*	FastMVE2()	**contour_plot_ind()**	weighted_mean()	*self_inner()*	mnist_data()	glove_experiments()
	ind2labels()			geometric_median()	*self_outer()*		fasttext_experiments()
			#Misc Figures#	weighted_median()	*symmetrize()*	#MNIST Manipulation#	code_experiment()
	#Evaluation Metrics#		**contour_plot()**			*flatten2d()*	weighting_kernel_graph()
	diagnostics()		**heatmap_plot()**	#Distances#	#RNG Manipulation	*faltten1d()*	
	diagnostics2D()		**component_plot()**	data2dist()	*get_seed()*	*mnist_mean()*	
	get_all_eval()		**bshape_plot()**	*mahalanobis2()*	*next_seed()*		
	mcc()			*mahalanobis1()*		#Word Embeddings#	
			#Auxiliary Functions#		#Auxiliary Functions#	word2vec_data_scratch()	
	#Auxiliary Functions#		*method_title()*	#Categorical Encodings#	*text_subscript()*	glove_data_scratch()	
	confusion_matrix()		*fast_contour_plot()*	vec_cat_encode()	*ts()*	fasttext_data_scratch()	
	reduce_mat()		*get_default_colour()*	mat_cat_encode()	*duplicates()*	*get_embedding()*	
			get_different_colour()			*get_embeddings()*	
			bshape()	#Data Transformations#			
				transform_loc()			
				transform_01()			
				transform_z()			
				transform_rob()			
				#Used for gpca/ics#			
				fix_signs!()			

To do

- Check the code.
- Put it as a package.

All contributors are welcome.

Outline of the presentation

Invariant coordinate selection

ICS - methodology

ICS - implementations

Conclusion

Tutorial

Conclusion

ICS is an attractive unsupervised multivariate method:

- Designed to find structure in a low-dimensional subspace:
  - for outlier detection,
  - as pre-processing for clustering
- Affine invariant.

Conclusion

ICS is an attractive unsupervised multivariate method:

- Designed to find structure in a low-dimensional subspace:
  - for outlier detection,
  - as pre-processing for clustering
- Affine invariant.

Still presents some challenges:

- Choice of scatter matrices.
- Choice of components.

Some perspectives

Implementation:

- R ecosystem to maintain and improve:

  ICS, **ICSShiny**, ICtest, ICSOutlier, ICSClust.
  Add algorithms for functional, compositional, collinear or HDLSS data.
- Python (work in progress)
- Julia (work in progress)

Some perspectives

Implementation:

- R ecosystem to maintain and improve:

  ICS, **ICSShiny**, ICtest, ICSOutlier, ICSClust.

  Add algorithms for functional, compositional, collinear or HDLSS data.
- Python (work in progress)
- Julia (work in progress)

Github repository:

- https://github.com/AuroreAA/ICS-implementation

All contributors are welcome.

Outline of the presentation

Invariant coordinate selection

ICS - methodology

ICS - implementations

Conclusion

Tutorial

References I

Alashwali, F. and Kent, J. T. (2016). The use of a common location measure in the invariant coordinate selection and projection pursuit. Journal of Multivariate Analysis, 152:145–161.

Alfons, A., Archimbaud, A., Nordhausen, K., and Ruiz-Gazen, A. (2024). Tandem clustering with invariant coordinate selection. Econometrics and Statistics.

Archimbaud, A., Alfons, A., Nordhausen, K., and Ruiz-Gazen, A. (2023a). ICSClust: Tandem Clustering with Invariant Coordinate Selection. R package version 0.1.0.

Archimbaud, A., Boulfani, F., Gendre, X., Nordhausen, K., Ruiz-Gazen, A., and Virta, J. (2025a). ICS for multivariate functional anomaly detection with applications to predictive maintenance and quality control. Econometrics and Statistics, 33:282–303.

Archimbaud, A., Drmač, Z., Nordhausen, K., Radojičić, U., and Ruiz-Gazen, A. (2023b). Numerical considerations and a new implementation for ICS. SIAM Journal on Mathematics of Data Science (SIMODS), 5(1):97–121.

Archimbaud, A., May, J., Nordhausen, K., and Ruiz-Gazen, A. (2025b). ICSShiny: ICS via a Shiny Application. R package version 0.6.

Archimbaud, A., Nordhausen, K., and Ruiz-Gazen, A. (2018a). ICS for multivariate outlier detection with application to quality control. Computational Statistics & Data Analysis, 128:184–199.

Archimbaud, A., Nordhausen, K., and Ruiz-Gazen, A. (2018b). ICSOutlier: Unsupervised outlier detection for low-dimensional contamination structure. The R Journal, 10(1):234–250.

Arias, V. B., Garrido, L., Jenaro, C., Martínez-Molina, A., and Arias, B. (2020). A little garbage in, lots of garbage out: Assessing the impact of careless responding in personality survey data. Behavior Research Methods, 52:2489–2505.

Becquart, C., Archimbaud, A., Ruiz-Gazen, A., Prilć, L., and Nordhausen, K. (2024). Invariant coordinate selection and fisher discriminant subspace beyond the case of two groups. arXiv preprint arXiv:2409.17631.

Cardoso, J.-F. (1989). Source separation using higher order moments. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pages 2109–2112. IEEE.

References II

Caussinus, H., Fekri, M., Hakam, S., and Ruiz-Gazen, A. (2003). A monitoring display of multivariate outliers. Computational Statistics & Data Analysis, 44(1):237–252.

Caussinus, H. and Ruiz-Gazen, A. (1990). Interesting projections of multidimensional data by means of generalized principal component analyses. In Compstat, pages 121–126. Springer.

Caussinus, H. and Ruiz-Gazen, A. (1993a). Projection pursuit and generalized principal component analysis. In Morgenthaler, S., Ronchetti, E., and Stahel, W. A., editors, New directions in statistical data analysis and robustness., Monte Verita, Proceedings of the Centro Stefano Franciscini Ascona Series. Springer.

Caussinus, H. and Ruiz-Gazen, A. (1993b). Projection pursuit and generalized principal component analysis. New Directions in Statistical Data Analysis and Robustness, pages 35–46.

Caussinus, H. and Ruiz-Gazen, A. (1995). Metrics for finding typical structures by means of principal component analysis. In Data Sci. and its Applications, pages 177–192, Japan. Harcourt Brace.

Caussinus, H. and Ruiz-Gazen, A. (2007). Classification and generalized principal component analysis. In Selected contributions in data analysis and classification, pages 539–548. Springer.

Claassen, C. (2023). Generalising invariant coordinate selection to a non-linear dimensionality reduction method. Master's thesis.

Dray, S. (2008). On the number of principal components: A test of dimensionality based on measurements of similarity between matrices. Computational Statistics and Data Analysis, 52(4):2228 – 2237.

Dümbgen, L., Gysel, K., and Perler, F. (2021). Refining invariant coordinate selection via local projection pursuit. arXiv preprint arXiv:2112.11998.

Fekri, M. and Ruiz-Gazen, A. (2015). A B-robust non-iterative scatter matrix estimator: Asymptotics and application to cluster detection using invariant coordinate selection. In Nordhausen, K. and Taskinen, S., editors, Modern Nonparametric, Robust and Multivariate Methods: Festschrift in Honour of Hannu Oja, pages 395–423. Springer International Publishing, Cham.

References III

Fischer, D., Honkatukia, M., Tuiskula-Haavisto, M., Nordhausen, K., Cavero, D., Preisinger, R., and Vilkki, J. (2017). Subgroup detection in genotype data using invariant coordinate selection. BMC Bioinformatics, 18:173–181.

Fischer, D., Nordhausen, K., and Oja, H. (2020). On linear dimension reduction based on diagonalization of scatter matrices for bioinformatics downstream analyses. Heliyon, 6:e05732.

Goldberg, L. R. (1992). The development of markers for the big-five factor structure. Psychological assessment, 4(1):26.

Hennig, C. (2009). Discussion of “Invariant Co-ordinate Selection”, by D. E. Tyler, F. Critchley, L. Dümbgen, and H. Oja. Journal of the Royal Statistical Society B, 71:579–583.

Luo, W. and Li, B. (2016). Combining eigenvalues and variation of eigenvectors for order determination. Biometrika, 103(4):875–887.

Luo, W. and Li, B. (2021). On order determination by predictor augmentation. Biometrika, 108:557–574.

Nordhausen, K., Alfons, A., Archimbaud, A., Oja, H., Ruiz-Gazen, A., and Tyler, D. E. (2025). Tools for Exploring Multivariate Data: The Package ICS. R package version 1.4-2.

Nordhausen, K., Oja, H., and Tyler, D. E. (2022). Asymptotic and Bootstrap Tests for Subspace Dimension. Journal of Multivariate Analysis, 188:104830.

Nordhausen, K., Oja, H., Tyler, D. E., and Virta, J. (2017). Asymptotic and bootstrap tests for the dimension of the non-gaussian subspace. IEEE Signal Processing Letters, 24(6):887–891.

Nordhausen, K. and Ruiz-Gazen, A. (2021). On the usage of joint diagonalization in multivariate statistics. Journal of Multivariate Analysis, page 104844.

Nordhausen, K. and Virta, J. (2019). An overview of properties and extensions of FOBI. Knowledge-Based Systems, 173:113–116.

Peña, D., Prieto, F. J., and Viladomat, J. (2010). Eigenvectors of a kurtosis matrix as interesting directions to reveal cluster structure. Journal of Multivariate Analysis, 101(9):1995–2007.

References IV

Peres-Neto, P. R., Jackson, D. A., and Somers, K. M. (2005). How many principal components? stopping rules for determining the number of non-trivial axes revisited. Computational Statistics and Data Analysis, 49(4):974 – 997.

Radojicic, U. and Nordhausen, K. (2020). Non-gaussian component analysis: Testing the dimension of the signal subspace. In Maciak, M., Pesta, M., and Schindler, M., editors, Analytical Methods in Statistics. AMISTAT 2019, pages 101–123. Springer, Cham.

Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. Mathematical statistics and applications, 8(283-297):37.

Ruiz-Gazen, A. (1996). A very simple robust estimator of a dispersion matrix. Computational Statistics & Data Analysis, 21:149–162.

Tyler, D. E., Critchley, F., Dümbgen, L., and Oja, H. (2009). Invariant coordinate selection. Journal of the Royal Statistical Society: Series B, 71(3):549–592.

Notations

- $\mathbf{X}$ is a p-dimensional random vector, $F_{\mathbf{X}}$ its cumulative distribution function and $\mathbf{m}(F_{\mathbf{X}})$ an affine equivariant location estimator.
- $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ a p-variate dataset. $\mathbf{x}_1, \ldots, \mathbf{x}_n$ follow the same distribution as $\mathbf{X}$.

Notations

- $\mathbf{X}$ is a p-dimensional random vector, $F_{\mathbf{X}}$ its cumulative distribution function and $\mathbf{m}(F_{\mathbf{X}})$ an affine equivariant location estimator.
- $\mathbf{X}_n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$ a p-variate dataset. $\mathbf{x}_1, \ldots, \mathbf{x}_n$ follow the same distribution as $\mathbf{X}$.
- $\mathcal{P}_p$ is the set of all symmetric positive definite matrices of order p.
- A scatter functional is defined as a matrix $\mathbf{S}(F_{\mathbf{X}}) \in \mathcal{P}_p$, uniquely defined at $F_{\mathbf{X}}$, which is affine equivariant in the sense that:

$$\mathbf{S}(F_{\mathbf{AX}+\boldsymbol{\gamma}}) = \mathbf{AS}(F_{\mathbf{X}})\mathbf{A}',$$

for all $p \times p$ non-singular matrices $\mathbf{A}$ and all $\boldsymbol{\gamma} \in \mathbb{R}^p$.

For sake of simplicity, the dependence on $F_{\mathbf{X}}$ is dropped.

Choice of scatter matrices I

- COV – COV_4 or FOBI Peña et al. (2010); Cardoso (1989); Nordhausen and Virta (2019); Fischer et al. (2017, 2020)

$$\text{COV}_4(\boldsymbol{X}_n) = \frac{1}{n}\sum_{i=1}^{n} r^2(\boldsymbol{x}_i)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_n)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_n)^\top,$$

where $r^2(\boldsymbol{x}_i) = (\boldsymbol{x}_i - \bar{\mathbf{x}}_n)^\top \text{COV}(\boldsymbol{x}_n)^{-1}(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_n)$ is the squared Mahalanobis distance.

Choice of scatter matrices II

- $\mathrm{MCD}_\alpha - \mathrm{COV}$ and $\mathrm{RMCD}_\alpha - \mathrm{COV}$: (Reweighted) Minimum Covariance Determinant Rousseeuw (1985).

$$\mathrm{MCD}_\alpha(\boldsymbol{X}_n) = c_\alpha \frac{1}{n_\alpha} \sum_{j=1}^{n_\alpha} (\boldsymbol{x}_{i_j} - \bar{\boldsymbol{x}}_{\alpha,n})(\boldsymbol{x}_{i_j} - \bar{\boldsymbol{x}}_{\alpha,n})^\top,$$

where $n_\alpha = \lceil \alpha n \rceil$ observations for which the sample covariance matrix has the smallest determinant and usually $\alpha \in [0.5, 1]$.
In our ecosystem it is also interesting to consider values of α that are smaller than 0.5.

CHOICE OF SCATTER MATRICES III

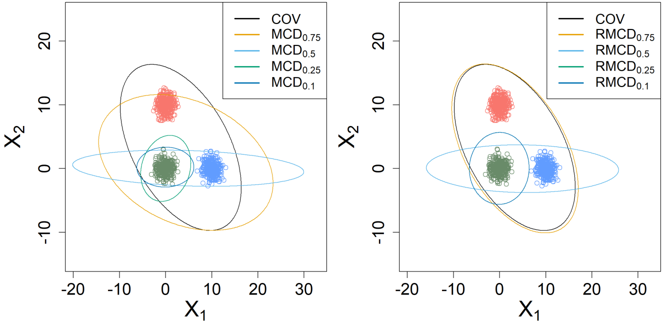


Figure: Shapes of different scatter matrices for a sample from a bivariate Gaussian mixture distribution with three balanced clusters.

Choice of scatter matrices IV

- SCOV – COV, TCOV – UCOV Caussinus and Ruiz-Gazen (1993a, 1995, 2007); Ruiz-Gazen (1996); Caussinus and Ruiz-Gazen (1990); Fekri and Ruiz-Gazen (2015)

Choice of scatter matrices IV

- SCOV − COV, TCOV − UCOV Caussinus and Ruiz-Gazen (1993a, 1995, 2007); Ruiz-Gazen (1996); Caussinus and Ruiz-Gazen (1990); Fekri and Ruiz-Gazen (2015) ⇒ TCOV − COV:

$$\mathrm{TCOV}_\beta(\boldsymbol{X}_n) = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(\beta\, r^2(\boldsymbol{x}_i, \boldsymbol{x}_j))(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(\beta, r^2(\boldsymbol{x}_i, \boldsymbol{x}_j))},$$

where $r^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \mathrm{COV}(\boldsymbol{x}_n)^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)$,
$w(x) = \exp(-x/2)$, $\beta = 4$.

Choice of scatter matrices IV

- SCOV – COV, TCOV – UCOV Caussinus and Ruiz-Gazen (1993a, 1995, 2007); Ruiz-Gazen (1996); Caussinus and Ruiz-Gazen (1990); Fekri and Ruiz-Gazen (2015) ⇒ TCOV – COV:

  $$\mathrm{TCOV}_\beta(\boldsymbol{X}_n) = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(\beta\, r^2(\boldsymbol{x}_i,\boldsymbol{x}_j))(\boldsymbol{x}_i-\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j)^\top}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w(\beta, r^2(\boldsymbol{x}_i,\boldsymbol{x}_j))},$$

  where $r^2(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\boldsymbol{x}_i-\boldsymbol{x}_j)^\top \mathrm{COV}(\boldsymbol{x}_n)^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j)$, $w(x) = \exp(-x/2)$, $\beta = 4$.

- LCOV – COV Hennig (2009), a local shape matrix based on the aggregation of covariance matrices computed based on 10% of the nearest neighbors (with regard to the Mahalanobis distance).

Parallel Analysis (PA)

As in Peres-Neto et al. (2005) for PCA or in Caussinus et al. (2003).

Computation of cut-offs:

- 10 000 simulations of $\mathcal{N}_n(0, I_p)$
- ICS with $\mathrm{COV}(\mathbf{X}_n)^{-1}\mathrm{COV}_4(\mathbf{X}_n)$
- Quantiles of the ICS eigenvalues at level $1 - \frac{\alpha}{i}$ for each component i, with $\alpha = 5\%$ and $i = 1, \ldots, p$.

Parallel Analysis (PA)

As in Peres-Neto et al. (2005) for PCA or in Caussinus et al. (2003).

Computation of cut-offs:

- 10 000 simulations of $\mathcal{N}_n(0, I_p)$
- ICS with $\mathrm{COV}(\mathbf{X}_n)^{-1}\mathrm{COV}_4(\mathbf{X}_n)$
- Quantiles of the ICS eigenvalues at level $1 - \frac{\alpha}{i}$ for each component i, with $\alpha = 5\%$ and $i = 1, \ldots, p$.

Test:

- Sequentially testing if the ICS eigenvalues are higher than corresponding quantiles.
- Stop as soon as one is lower than the cut-off.

Normality Tests

Finding the first i^{th} coordinate with no longer information about the structure of the data through normality tests.

Normality Tests

Finding the first i^{th} coordinate with no longer information about the structure of the data through normality tests.

Normality tests:

- The D'Agostino test of skewness (DA),
- The Anscombe-Glynn (AG) test of kurtosis,
- The Bonett-Seier (BS) test of Geary's kurtosis,
- The Jarque-Bera (JB) test for normality which is based on both skewness and kurtosis measures,
- The Shapiro-Wilk (SW) normality test.

Normality Tests

Finding the first i^{th} coordinate with no longer information about the structure of the data through normality tests.

Normality tests:

- The D'Agostino test of skewness (DA),
- The Anscombe-Glynn (AG) test of kurtosis,
- The Bonett-Seier (BS) test of Geary's kurtosis,
- The Jarque-Bera (JB) test for normality which is based on both skewness and kurtosis measures,
- The Shapiro-Wilk (SW) normality test.

Stopping rule

- Sequentially testing if the ICs are gaussian or not.
- Stop as soon as one is gaussian based on the corrected level of 5%.