# ITE (Information Theoretical Estimators) Matlab/Octave Toolbox Release 0.26

Zoltán Szabó

December 22, 2012

## Contents

**1 Introduction** **4**

**2 Installation** **6**

**3 Estimation of Information Theoretical Quantities** **10**
3.1 Base Estimators . . . 11
3.1.1 Entropy Estimators . . . 11
3.1.2 Mutual Information Estimators . . . 13
3.1.3 Divergence Estimators . . . 16
3.1.4 Association Estimators . . . 19
3.1.5 Cross Estimators . . . 21
3.2 Meta Estimators . . . 21
3.2.1 Entropy Estimators . . . 21
3.2.2 Mutual Information Estimators . . . 23
3.2.3 Divergence Estimators . . . 25
3.2.4 Association Estimators . . . 26
3.2.5 Cross Estimators . . . 27
3.3 Uniform Syntax of the Estimators . . . 27

**4 ITE Application in Independent Process Analysis (IPA)** **29**
4.1 IPA Models . . . 29
4.1.1 Independent Subspace Analysis (ISA) . . . 29
4.1.2 Extensions of ISA . . . 32
4.2 Estimation via ITE . . . 35
4.2.1 ISA . . . 36
4.2.2 Extensions of ISA . . . 38
4.3 Performance Measure, the Amari-index . . . 41
4.4 Dataset-, Model Generators . . . 42

**5 Directory Structure of the Package** **46**

**A Citation of the ITE Toolbox** **47**

**B Abbreviations** **47**

**C Functions with Octave-Specific Adaptations** **47**

**D Further Definitions** **47**

**E Estimation Formulas – Lookup Table** **49**
E.1 Entropy . . . 50
E.2 Mutual Information . . . 54
E.3 Divergence . . . 57
E.4 Association Measures . . . 60
E.5 Cross Quantities . . . 60

# List of Figures

1 IPA problem family, relations . . . 36
2 ISA demonstration . . . 42
3 Illustration of the datasets . . . 44

# List of Tables

1 External, dedicated packages increasing the efficiency of ITE . . . 11
2 Entropy estimators (base) . . . 13
3 Mutual information estimators (base) . . . 17
4 Divergence estimators (base) . . . 19
5 Association estimators (base) . . . 20
6 Cross estimators (base) . . . 21
7 Entropy estimators (meta) . . . 23
8 Mutual information estimators (meta) . . . 26
9 Divergence estimators (meta) . . . 27
10 Well-scaling approximation for the permutation search problem in ISA . . . 33
11 ISA formulations . . . 37
12 Optimizers for unknown subspace dimensions, spectral clustering method . . . 37
13 Optimizers for given subspace dimensions, greedy method . . . 38
14 Optimizers for given subspace dimensions, cross-entropy method . . . 38
15 Optimizers for given subspace dimensions, exhaustive method . . . 38
16 IPA separation principles . . . 40
17 IPA subtasks and estimators . . . 40
18 k-nearest neighbor methods . . . 41
19 Minimum spanning tree methods . . . 41
20 IPA model generators . . . 44
21 Description of the datasets . . . 45
22 Generators of the datasets . . . 45
23 Abbrevations . . . 48
24 Functions with Octave-specific adaptations . . . 49

# List of Examples

1 ITE installation (output; with compilation) . . . 10
2 Entropy estimation (base-1: usage) . . . 11
3 Entropy estimation (base-2: usage) . . . 12
4 Mutual information estimation (base: usage) . . . 16
5 Divergence estimation (base: usage) . . . 18
6 Association estimation (base: usage) . . . 20
7 Cross estimation (base: usage) . . . 21
8 Entropy estimation (meta: initialization) . . . 21
9 Entropy estimation (meta: estimation) . . . 22
10 Entropy estimation (meta: usage) . . . 22
11 Mutual information estimator (meta: initialization) . . . 24
12 Mutual information estimator (meta: estimation) . . . 24

13 Mutual information estimator (meta: usage) . . . 24
14 Divergence estimator (meta: initialization) . . . 25
15 Divergence estimator (meta: estimation) . . . 26
16 Divergence estimator (meta: usage) . . . 26
17 Entropy estimation (high-level, usage) . . . 29
18 ISA-1 . . . 36
19 ISA-2 . . . 36
20 ISA-3 . . . 37

# List of Templates

1 Entropy estimator: initialization . . . 27
2 Mutual information estimator: initialization . . . 27
3 Divergence estimator: initialization . . . 27
4 Association estimator: initialization . . . 27
5 Cross estimator: initialization . . . 27
6 Entropy estimator: estimation . . . 28
7 Mutual information estimator: estimation . . . 28
8 Divergence estimator: estimation . . . 28
9 Association estimator: estimation . . . 28
10 Cross estimator: estimation . . . 28

# 1 Introduction

Since the pioneering work of Shannon [79], *entropy, mutual information, divergence* measures and their extensions have found a broad range of applications in many areas of machine learning. Entropies provide a natural notion to quantify the *uncertainty* of random variables, mutual information type indices measure the *dependence* among its arguments, divergences offer efficient tools to define the 'distance' of probability measures. Particularly, in the classical Shannon case, these three concepts form a gradually widening chain: entropy is equal to the self mutual information of a random variable, mutual information is identical to the divergence of the joint distribution and the product of the marginals [14]. Applications of Shannon entropy, -mutual information, -divergence and their generalizations cover, for example, (i) feature selection, (ii) clustering, (iii) independent component/subspace analysis, (iii) image registration, (iv) boosting, (v) optimal experiment design, (vi) causality detection, (vii) hypothesis testing, (viii) Bayesian active learning, (ix) structure learning in graphical models, (x) region-of-interest tracking, among many others. For an excellent review on the topic, the reader is referred to [6, 112, 109, 5, 59].

Independent component analysis (ICA) [37, 9, 10] a central problem of signal processing and its generalizations can be formulated as optimization problems of information theoretical objectives. One can think of ICA as a cocktail party problem: we have some speakers (sources) and some microphones (sensors), which measure the mixed signals emitted by the sources. The task is to estimate the original sources from the mixed observations only. Traditional ICA algorithms are one-dimensional in the sense that all sources are assumed to be *independent* real valued random variables. However, many important applications underpin the relevance of considering extensions of ICA, such as the independent subspace analysis (ISA) problem [8, 15]. In ISA, the independent sources can be multidimensional: we have a cocktail-party, where more than one *group* of musicians are playing at the party. Successful applications of ISA include (i) the processing of EEG-fMRI, ECG data and natural images, (ii) gene expression analysis, (iii) learning of face view-subspaces, (iv) motion segmentation, (v) single-channel source separation, (vi) texture classification, (vii) action recognition in movies.

One of the most relevant and fundamental hypotheses of the ICA research is the ISA separation principle [8]: the ISA task can be solved by ICA followed by clustering of the ICA elements. This principle (i) forms the basis of the state-of-the-art ISA algorithms, (ii) can be used to design algorithms that scale well and efficiently estimate the dimensions of the hidden sources, (iii) has been recently proved [93] and (iv) can be extended to different linear-, controlled-, post nonlinear-, complex valued-, partially observed models, as well as to systems with nonparametric source dynamics. For a recent review on the topic, see [96].

Although there exist many exciting applications of information theoretical measures, to the best of our knowledge, available packages in this domain focus on (i) discrete variables, or (ii) quite specialized applications and information theoretical estimation methods. Our **goal** is to fill this serious gap by coming up with a (i) highly modular, (ii) free and open source, (iii) multi-platform toolbox, the ITE (information theoretical estimators) package, which

1. is capable of estimating *many* different variants of entropy, mutual information, divergence measures, and related association-, cross quantities:

   - entropy: Shannon entropy, Rényi entropy, Tsallis entropy, complex entropy,
   - mutual information: generalized variance (GV), kernel canonical correlation analysis (KCCA), kernel generalized variance (KGV), Hilbert-Schmidt independence criterion (HSIC), Shannon mutual information, $L_2$ mutual information, Rényi mutual information, Tsallis mutual information, copula-based kernel dependency, multivariate version of Hoeffding's $\Phi$, Schweizer-Wolff's $\sigma$ and $\kappa$, complex mutual information, Cauchy-Schwartz quadratic mutual information (QMI), Euclidean distance based QMI, distance covariance, distance correlation,
   - divergence: Kullback-Leibler divergence, $L_2$ divergence, Rényi divergence, Tsallis divergence Hellinger distance, Bhattacharyya distance, maximum mean discrepancy (MMD; integral probability metric), J-distance, Cauchy-Schwartz divergence, Euclidean distance based divergence, energy distance (specially the Cramer-Von Mises distance),
   - association measures: multivariate extensions of Spearman's $\rho$,
   - cross quantities: cross-entropy,

   based on

   - nonparametric methods[1]: k-nearest neighbors, generalized k-nearest neighbors, weighted k-nearest neighbors,

[1] It is highly advantageous to apply nonparametric approaches to estimate information theoretical quantities. The bottleneck of the 'opposite' plug-in type methods, which estimate the underlying density and then plug it in into the appropriate integral formula, is that the unknown densities are nuisance parameters. As a result, plug-in type estimators scale poorly as the dimension is increasing.

minimum spanning trees, geodesic spanning forests, random projection, kernel techniques, ensemble methods, sample spacing,

- kernel density estimation (KDE): in plug-in scheme.

2. offers a *simple and unified framework* to
   (a) easily construct new estimators from existing ones or from scratch, and
   (b) transparently use the obtained estimators in information theoretical optimization problems.
3. with a *prototype application* in ISA and its extensions including
   - 6 different ISA objectives,
   - 4 optimization methods: (i) handling known and unknown subspace dimensions as well, with (ii) further objective-specific accelerations,
   - 5 extended problem directions: (i) different linear-, (ii) controlled-, (iii) post nonlinear-, (iii) complex valued-, (iv) partially observed models, (v) as well as systems with nonparametric source dynamics; which can be used in combinations as well.

The technical details of the ITE package are as follows:

- **Author**: Zoltán Szabó.
  - Homepage: `http://nipg.inf.elte.hu/szzoli`
  - Email: `szzoli@cs.elte.hu`
  - Affiliation: Eötvös Loránd University, Faculty of Informatics (Computer Science), Pázmány Péter sétány 1/C, Budapest, H-1117, Hungary.
- **Documentation of the source**: the source code of ITE has been enriched with numerous comments, examples, and pointers where the interested user can find further mathematical details about the embodied techniques.
- **License** (GNU GPLv3 or later): ITE is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with ITE. If not, see <http://www.gnu.org/licenses/>.
- **Citing**: If you use the ITE toolbox in your work, please cite the papers [93, 96] (.bib in Appendix A).
- **Platforms**: The ITE package has been extensively tested on Windows and Linux. However, since it is made of standard Matlab/Octave and C/C++ files, it is expected to work on alternative platforms as well.
- **Environments**: Matlab[2], Octave[3].
- **Requirements**: The ITE package is self-contained, it only needs
  - a Matlab or an Octave environment with standard toolboxes:
    - Matlab: Image Processing, Optimization, Statistics.
    - Octave[4]: Image Processing (image), Statistics (statistics), Input/Output (io, required by statistics), Ordinary Differential Equations (odepkg), Bindings to the GNU Scientific Library (gsl), ANN wrapper (ann).
  - a C/C++ compiler – if you would like to further speed up the computations.
- **Comments, feedbacks**: are welcome.
- **Homepage of the ITE toolbox**: `https://bitbucket.org/szzoli/ite/`
- **Follower**: become a follower to be always up-to-date with ITE (`https://bitbucket.org/szzoli/ite/follow`).

[2] `http://www.mathworks.com/products/matlab/`
[3] `http://www.gnu.org/software/octave/`
[4] See `http://octave.sourceforge.net/packages.php`.

The remainder of this document is organized as follows:

- Section 2 is about the installation of the ITE package. Section 3 focuses on the estimation of information theoretical quantities (entropy, mutual information, divergence, association and cross measures) and their realization in ITE. In Section 4, we present an application of Section 3 included in the ITE toolbox. The application considers the extension of independent subspace analysis (ISA, independent component analysis with multidimensional sources) to different linear-, controlled-, post nonlinear-, complex valued-, partially observed problems, as well as problems dealing with nonparametric source dynamics, i.e., the independent process analysis (IPA) problem family. Section 5 is about the organization of the directories of the ITE toolbox.
- Citing information of the ITE package is provided in Appendix A. Abbreviations of the paper are listed in Appendix B (Table 23). Functions with Octave-specific adaptations are summarized in Appendix C (Table 24). Some further formal definitions (measure of concordance, semimetric space of negative type) are given in Appendix D to make the documentation self-contained. A brief summary (lookup table) of the underlying entropy, mutual information, divergence, association and cross measure computations can be found in Appendix E.

# 2 Installation

This section is about (i) the installation of the ITE toolbox, and (ii) the external packages, dedicated solvers embedded in the ITE package. The purpose of this inclusion is twofold:

- to further increase the efficiency of certain subtasks to be solved (e.g., k-nearest neighbor search, finding minimum spanning trees, some subtasks revived by the IPA separation principles (see Section 4.1)),
- to provide both purely Matlab/Octave implementations, and specialized (often faster) non-Matlab/-Octave solutions that can be called from Matlab/Octave.

The core of the ITE toolbox has been written in Matlab, as far it was possible in an Octave compatible way. The particularities of Octave has been taken into account by adapting the code to the *actual* environment (Matlab/Octave). The working environment can be queried (e.g., in case of extending the package it is also useful) by the `working_environment_Matlab.m` function included in ITE. Adaptations has been carried out in the functions listed in Appendix C (Table 24). The functionalities extended by the external packages are also available in both environments (Table 1).

Here, a short description of the embedded/downloaded packages (directory 'shared/embedded', 'shared/downloaded') is given:

1. **fastICA** (directory 'shared/embedded/FastICA'; version 2.5):
   - **URL**: `http://research.ics.tkk.fi/ica/fastica/`
   - **License**: GNU GPLv2 or later.
   - **Solver**: ICA (independent component analysis).
   - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
   - **Environment**: Matlab, Octave.
   - **Note**: By commenting out the `g_FastICA_interrupt` variable in `fpica.m`, the `fastica.m` function can be used in Octave, too. The provided fastICA code in the ITE toolbox contains this modification.
2. **Complex fastICA** (directory 'shared/embedded/CFastICA')
   - **URL**: `http://www.cs.helsinki.fi/u/ebingham/software.html`, `http://users.ics.aalto.fi/ella/publications/cfastica_public.m`
   - **License**: GNU GPLv2 or later.
   - **Solver**: complex ICA.
   - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
   - **Environment**: Matlab, Octave.
3. **ANN (approximate nearest neighbor) Matlab wrapper** (directory 'shared/embedded/ann_wrapperM'; version 'Mar2012'):
   - **URL**: `http://www.wisdom.weizmann.ac.il/~bagon/matlab.html`, `http://www.wisdom.weizmann.ac.il/~bagon/matlab_code/ann_wrapper_Mar2012.tar.gz`
   - **License**: GNU LGPLv3.

- **Solver**: approximate nearest neighbor computation.
- **Installation**: Follow the instructions in the ANN wrapper package (README.txt: INSTALLATION) till 'ann_class_compile'. Note: If you use a more recent C++ compiler (e.g., g++ on Linux), you have to include the following 2 lines into the original code to be able to compile the source:
  (a) '#include <cstdlib>' to 'ANNx.h'
  (b) '#include <cstring>' to 'kd_tree.h'
  The provided ANN code in the ITE package contains these modifications.
- **Environment**: Matlab, Octave[5].
- **Note**: fast nearest neighbor alternative of `knnsearch` ∈ Matlab: Statistics Toolbox.

4. **MatlabBGL** (directory 'shared/embedded/MatlabBGL', version 4.0)
   - **URL**: `https://github.com/dgleich/matlab-bgl`, `http://www.mathworks.com/matlabcentral/fileexchange/10922`
   - **License**: 2-clause BSD, and GNU GPLv2 or later.
   - **Solver**: minimum spanning trees: Prim and Kruskal algorithm.
   - **Installation**: Add it with subfolders to your Matlab/Octave PATH. Note:
     - The package includes precompiled MEX files for Windows (32-bit and 64-bit), and Linux (32-bit and 64-bit for Matlab 2006b+), and MacOSX (32-bit Intel and 32-bit PPC).
     - The package includes source code to compile on other platforms as well.
   - **Environment**: Matlab, Octave[6].
   - **Note**: alternative of '14) = pmtk3' in finding minimum spanning trees.

5. **FastKICA** (directory 'shared/embedded/FastKICA', version 1.0):
   - **URL**: `http://people.kyb.tuebingen.mpg.de/arthur/fastkica.htm`
   - **License**: GNU GPL v2 or later.
   - **Solver**: HSIC (Hilbert-Schmidt independence criterion) mutual information estimator.
   - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
   - **Environment**: Matlab, Octave.
   - **Note**: one can extend the implementation of HSIC to measure the dependence of $d_m$-dimensional variables, too. The ITE toolbox contains this modification.

6. **NCut** (Normalized Cut, directory 'shared/embedded/NCut'; version 9):
   - **URL**: `http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html`, `http://www.seas.upenn.edu/~timothee/software/ncut/Ncut_9.zip`
   - **License**: GNU GPLv3.
   - **Solver**: spectral clustering, fixed number of groups.
   - **Installation**: Run `compileDir_simple.m` from Matlab to the provided directory of functions.
   - **Environment**: Matlab.
   - **Note**: the package is a fast alternative of '11) = spectral clustering'.

7. **sqdistance** (directory 'shared/embedded/sqdistance')
   - **URL**: `http://www.mathworks.com/matlabcentral/fileexchange/24599-pairwise-distance-matrix/`, `http://www.mathworks.com/matlabcentral/fileexchange/24599-pairwise-distance-matrix?download=true`
   - **License**: 2-clause BSD.
   - **Solver**: fast pairwise distance computation.
   - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
   - **Environment**: Matlab, Octave.
   - **Note**: compares favourably to the Matlab/Octave function `pdist`.

8. **TCA** (directory 'shared/embedded/TCA'; version 1.0):

[5] At the time of writing this paper, the Octave ANN wrapper (`http://octave.sourceforge.net/ann/index.html`, version 1.0.2) supports 2.9.12 ≤ Octave < 3.4.0. According to our experiences, however the ann wrapper can also be used for higher versions of Octave provided that (i) a new swig package (`www.swig.org/`) is used (>=2.0.5), (ii) a new 'SWIG=swig' line is inserted in src/ann/bindings/Makefile (the ITE package contains the modified makefile), and (iii) the row containing 'typedef OCTAVE_IDX_TYPE octave_idx_type;' (in '.../octave/config.h') is commented out for the time of 'make'-ing.

[6] With some trick, the MatlabBGL works on Octave, see `https://answers.launchpad.net/matlab-bgl/+question/48686`.

- **URL**: `http://www.di.ens.fr/~fbach/tca/index.htm`, `http://www.di.ens.fr/~fbach/tca/tca1_0.tar.gz`
- **License**: GNU GPLv2 or later.
- **Solver**: KCCA (kernel canonical correlation analysis) / KGV (kernel generalized variance) estimator, incomplete Cholesky decomposition.
- **Installation**: Add it with subfolders to your Matlab/Octave PATH.
- **Environment**: Matlab, Octave.
- **Note**: Incomplete Cholesky factorization can be carried out by the Matlab/Octave function `chol_gauss.m`. One can also compile the included `chol_gauss.c` to attain improved performance. Functions provided in the ITE toolbox contain extensions of the KCCA and KGV indices to measure the dependence of $d_m$-dimensional variables. The computations have also been accelerated in ITE by '7) = sqdistance'.

9. **Weighted kNN** (kNN: k-nearest neighbor; directory 'shared/embedded/weightedkNN' and the core of `HRenyi_weightedkNN_estimation.m`):
   - **URL**: `http://www-personal.umich.edu/~kksreddy/`
   - **License**: GNU GPLv3 or later.
   - **Solver**: Rényi entropy estimator based on the weighted k-nearest neighbor method.
   - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
   - **Environment**: Matlab, Octave.
   - **Note**: in the weighted kNN technique the weights are optimized. Since Matlab and Octave rely on different optimization engines, one has to adapt the weight estimation procedure to Octave. The `calculateweight.m` function in ITE contains this modification.

10. **E4** (directory 'shared/embedded/E4'):
    - **URL**: `http://www.ucm.es/info/icae/e4/`, `http://www.ucm.es/info/icae/e4/downfiles/E4.zip`
    - **License**: GNU GPLv2 or later.
    - **Solver**: AR (autoregressive) fit.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH[7].
    - **Environment**: Matlab, Octave.
    - **Note**: alternative of '13) = ARfit' in AR identification.

11. **spectral clustering** (directory 'shared/embedded/sp_clustering'):
    - **URL**: `http://www.mathworks.com/matlabcentral/fileexchange/34412-fast-and-efficient-spectral-clustering`
    - **License**: 2-clause BSD.
    - **Solver**: spectral clustering.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
    - **Note**: the package is a purely Matlab/Octave alternative of '6)=NCut'. It is advisable to alter the eigensystem computation in the `SpectralClustering.m` function to work stably in Octave; the modification is included in the ITE toolbox and is activated in case of Octave environment.

12. **clinep** (directory 'shared/embedded/clinep'):
    - **URL**: `http://www.mathworks.com/matlabcentral/fileexchange/8597-plot-3d-color-line/content/clinep.m`
    - **License**: 2-clause BSD.
    - **Solver**: Plots a 3D line with color encoding along the length using the patch function.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
    - **Note**: (i) calling of the cylinder function (in `clinep.m`) has to modified somewhat to work in Octave, and (ii) since 'gnuplot (as of v4.2) only supports 3D filled triangular patches' one has to use the fltk graphics toolkit in Octave for drawing. The included `cline.m` code in the ITE package contains these modifications.

13. **ARfit** (directory 'shared/downloaded/ARfit', version 'March 20, 2011')

[7] In Octave, this step results in a 'warning: function `.../shared/embedded/E4/vech.m` shadows a core library function'; it is OK, the two functions compute the same quantity.

- **URL**: `http://www.gps.caltech.edu/~tapio/arfit/`, `http://www.gps.caltech.edu/~tapio/arfit/arfit.zip`. Note: temporarily this website seems to be unavailable. The download link (at the moment) is `http://www.mathworks.com/matlabcentral/fileexchange/174-arfit?download=true`.
- **License**: ACM.
- **Solver**: AR identification.
- **Installation**: Download, extract and add it with subfolders to your Matlab/Octave PATH.
- **Environment**: Matlab, Octave.
- **Note**: alternative of '10) = E4' in AR identification.

14. **pmtk3** (directory 'shared/embedded/pmtk3', version 'Jan 2012')
    - **URL**: `http://code.google.com/p/pmtk3`, `http://code.google.com/p/pmtk3/downloads/detail?name=pmtk3-3jan11.zip&can=2&q=`.
    - **License**: MIT.
    - **Solver**: minimum spanning trees: Prim algorithm.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
    - **Note**: purely Matlab/Octave alternative of '4) = MatlabBGL' in finding minimum spanning trees.

15. **knn** (directory 'shared/embedded/knn', version 'Nov 02, 2010')
    - **URL**: `http://www.mathworks.com/matlabcentral/fileexchange/28897-k-nearest-neighbor-search`, `http://www.mathworks.com/matlabcentral/fileexchange/28897-k-nearest-neighbor-search?download=true`
    - **License**: 2-clause BSD.
    - **Solver**: kNN search.
    - **Installation**: Run the included `build` command to compile the partial sorting function `top.cpp`. Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
    - **Note**: Alternative of '3)=ANN' in finding k-nearest neighbors.

16. **SWICA** (directory 'shared/embedded/SWICA')
    - **URL**: `http://www.stat.purdue.edu/~skirshne/SWICA`, `http://www.stat.purdue.edu/~skirshne/SWICA/swica.tar.gz`
    - **License**: 3-clause BSD.
    - **Solver**: Schweizer-Wolff's $\sigma$ and $\kappa$ estimation.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
    - **Note**: one can also compile the included `SW_kappa.cpp` and `SW_sigma.cpp` functions to further accelerate computations (see '`build_SWICA.m`').

17. **ITL** (directory 'shared/embedded/ITL'; version '14.11.2012'):
    - **URL**: `http://www.sohanseth.com/ITL%20Toolbox.zip?attredirects=0`, `http://www.sohanseth.com/Home/codes`.
    - **License**: GNU GPLv3.
    - **Solver**: Cauchy-Schwartz quadratic mutual information, Euclidean distance based quadratic mutual information; and associated divergences.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.

A short summary of the packages can be found in Table 1. To ease installation, the ITE package contains an installation script, `ITE_install.m`. A typical usage is to cd to the directory 'code' and call `ITE_install(pwd)`. Running the script from Matlab/Octave, it (i) adds the main ITE directory with subfolders to the Matlab/Octave PATH, (ii) downloads and extracts the ARfit package, and (iii) compiles the embedded ANN, NCut, TCA, SWICA, knn packages, .cpp accelerations of the Hoeffding's $\Phi$ [see Eq. (19)], Edgeworth expansion based entropy [see Eq.(164)] computation, and the continuously differentiable sample spacing (CDSS) based estimator [see Eq. (193)].[8] The `ITE_install.m` script automatically detects

[8] The ITE package also offers purely Matlab/Octave implementations for the computation of Hoeffding's $\Phi$, Edgeworth expansion based entropy approximation and CDSS. Without compilation, these Matlab/Octave implementations are evoked.

the working environment (Matlab/Octave) and performs the installation accordingly, for example, it deletes the ann wrapper not suitable for the current working environment. The output of a successful installation in Matlab is given below (the Octave output is similar):

**Example 1 (ITE installation (output; with compilation))**

```
>> ITE_install(pwd); %after cd-ing to the code directory
Installation: started.
We are working in Matlab environment. => ann_wrapper for Octave: deleted.
ARfit package: downloading, extraction: started.
ARfit package: downloading, extraction: ready.
ITE directory: added with subfolders to the Matlab PATH.
ANN compilation: started.
ANN compilation: ready.
NCut compilation: started.
NCut compilation: ready.
TCA (chol_gauss.c) compilation: started.
TCA (chol_gauss.c) compilation: ready.
SWICA (SW_kappa.cpp, SW_sigma.cpp) compilation: started.
SWICA (SW_kappa.cpp, SW_sigma.cpp) compilation: ready.
Hoeffding_term1.cpp compilation: started.
Hoeffding_term1.cpp compilation: ready.
Edgeworth_t1_t2_t3.cpp compilation: started.
Edgeworth_t1_t2_t3.cpp compilation: ready.
compute_CDSS.cpp compilation: started.
compute_CDSS.cpp compilation: ready.
knn (top.cpp) compilation: started.
knn (top.cpp) compilation: ready.
-------------------
Installation tests:
ANN quick test: successful.
NCut quick test: successful.
ARfit quick test: successful.
knn quick test: successful.
```

# 3 Estimation of Information Theoretical Quantities

In this section we focus on the estimation of information theoretical quantities. Particularly, in the sequel, the underlying idea how the estimators are implemented in ITE are detailed, accompanied with definitions, numerous examples and extension possibilities/instructions.

The ITE package supports the estimation of many different variants of entropy, mutual information, divergence, association and cross measures:

1. From construction point of view, we distinguish two types of estimators in ITE: *base* (Section 3.1) and *meta* (Section 3.2) ones. Meta estimators are *derived* from existing base/meta ones by taking into account information theoretical identities. For example, by considering the well-known

$$I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = \sum_{m=1}^{M} H\left(\mathbf{y}^m\right) - H\left(\left[\mathbf{y}^1;\ldots;\mathbf{y}^M\right]\right) \tag{1}$$

   relation [14], one can estimate mutual information ($I$) by making use of existing entropy estimators ($H$).

2. From calling point of view, base and meta estimations follow exactly the same syntax (Section 3.3).

This modular implementation of the ITE package, makes it possible to

1. construct new estimators from existing ones, and

| Task | Package | Written in | Environment | Directory |
|---|---|---|---|---|
| ICA | fastICA | Matlab | Matlab, Octave | shared/embedded/FastICA |
| complex ICA | complex fastICA | Matlab | Matlab, Octave | shared/embedded/CFastICA |
| kNN search | ANN | C++ | Matlab | shared/embedded/ann_wrapperM[a] |
| kNN search | ANN | C++ | Octave[b] | shared/embedded/ann_wrapperO[a] |
| Prim-, Kruskal algorithm | MatlabBGL | C++ | Matlab, Octave[c] | shared/embedded/MatlabBGL |
| HSIC estimation | FastKICA | Matlab | Matlab, Octave | shared/embedded/FastKICA |
| spectral clustering | NCut | C++ | Matlab | shared/embedded/NCut |
| fast pairwise distance computation | sqdistance | Matlab | Matlab, Octave | shared/embedded/sqdistance |
| KCCA, KGV | TCA | Matlab, C | Matlab, Octave | shared/embedded/TCA |
| Rényi entropy via weighted kNNs | weighted kNN | Matlab | Matlab, Octave | shared/embedded/weightedkNN |
| AR fit | E4 | Matlab | Matlab, Octave | shared/embedded/E4 |
| spectral clustering | spectral clustering | Matlab | Matlab, Octave | shared/embedded/sp_clustering |
| trajectory plot | clinep | Matlab | Matlab, Octave | shared/embedded/clinep |
| AR fit | ARfit | Matlab | Matlab, Octave | shared/downloaded/ARfit |
| Prim algorithm | pmtk3 | Matlab | Matlab, Octave | shared/embedded/pmtk3 |
| kNN search | knn | Matlab, C++ | Matlab, Octave | shared/embedded/knn |
| Schweizer-Wolff's $\sigma$ and $\kappa$ | SWICA | Matlab, C++ | Matlab, Octave | shared/embedded/SWICA |
| QMIs + associated divergences | ITL | Matlab | Matlab, Octave | shared/embedded/ITL |

Table 1: External, dedicated packages increasing the efficiency of ITE.

[a] In 'ann_wrapperM' 'M' stands for Matlab, in 'ann_wrapperO' 'O' denotes Octave.
[b] See footnote 5.
[c] See footnote 6.

2. transparently use *any* of these estimators in information theoretical optimization problems (see Section 4) – provided that they follow a simple template described in Section 3.3.

## 3.1 Base Estimators

This section is about the *base* information theoretical estimators of the ITE package. Entropy estimation is in the focus of Section 3.1.1; in Section 3.1.2, Section 3.1.3, Section 3.1.4, Section 3.1.5 we consider mutual information, divergence, association and cross measure estimations, respectively.

### 3.1.1 Entropy Estimators

Let us start with a simple example: our goal is to estimate the Shannon entropy [79]

$$H(\mathbf{y}) = -\int_{\mathbb{R}^d} f(\mathbf{u}) \log f(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{2}$$

of a random variable $\mathbf{y} \in \mathbb{R}^d$ from which we have i.i.d. (independent identically distributed) samples $\{\mathbf{y}_t\}_{t=1}^T$, and $f$ denotes the density function of $\mathbf{y}$. The estimation of Shannon entropy can be carried out, e.g., by k-nearest neighbor techniques. Let us also assume that multiplicative contants are also important for us – in many applications, it is completely irrelevant whether we estimate, for example, $H(\mathbf{y})$ or $cH(\mathbf{y})$, where $c = c(d)$ is a constant depending only on the *dimension* of $\mathbf{y}$ ($d$), but *not on the distribution* of $\mathbf{y}$. By using the ITE package, the estimation can be carried out as simply as follows:

**Example 2 (Entropy estimation (base-1: usage))**

```
>Y = rand(5,1000);                          %generate the data of interest (d=5, T=1000)
>mult = 1;                                  %multiplicative constant is important
>co = HShannon_kNN_k_initialization(mult); %initialize the entropy ('H') estimator
                                            %('Shannon_kNN_k'), including the value of k
>H = HShannon_kNN_k_estimation(Y,co);       %perform entropy estimation
```

Alternative entropy measures of interest include the:

1. **Rényi entropy** [72]: defined as

$$H_{\mathrm{R},\alpha}(\mathbf{y}) = \frac{1}{1-\alpha} \log \int_{\mathbb{R}^d} f^\alpha(\mathbf{u}) \mathrm{d}\mathbf{u}, (\quad \alpha \neq 1) \tag{3}$$

where the random variable $\mathbf{y} \in \mathbb{R}^d$ have density function $f$. The Shannon entropy [Eq. (2)] is a special case of the Rényi entropy family, in limit:

$$\lim_{\alpha \to 1} H_{\mathrm{R},\alpha} = H. \tag{4}$$

2. **Tsallis entropy** (also called the Havrda and Charvát entropy) [106, 27]: closely related to the Rényi entropy, defined as

$$H_{\mathrm{T},\alpha}(\mathbf{y}) = \frac{1}{\alpha - 1} \left( 1 - \int_{\mathbb{R}^d} f^\alpha(\mathbf{u}) \mathrm{d}\mathbf{u} \right), \quad \alpha \neq 1. \tag{5}$$

The Shannon entropy is a special case of the Tsallis entropy family, in limit:

$$\lim_{\alpha \to 1} H_{\mathrm{T},\alpha} = H. \tag{6}$$

In the ITE toolbox, $H_{\mathrm{R},\alpha}$ and $H_{\mathrm{T},\alpha}$ can be estimated similarly to the Shannon entropy $H$ (see Example 2):

**Example 3 (Entropy estimation (base-2: usage))**

```
>Y = rand(5,1000);                          %generate the data of interest (d=5, T=1000)
>mult = 1;                                  %multiplicative constant is important
>co = HRenyi_kNN_k_initialization(mult);    %initialize the entropy ('H') estimator ('Renyi_kNN_k'),
                                            %including the value of k and α
>H = HRenyi_kNN_k_estimation(Y,co);         %perform entropy estimation
```

Beyond k-nearest neighbor based $H$ (see [41] ($S = \{1\}$), [81, 23] $S = \{k\}$; in ITE '`Shannon_kNN_k`') and $H_{\mathrm{R},\alpha}$ estimation methods [115, 45] ($S = \{k\}$; '`Renyi_kNN_k`'), the ITE package also provide functions for the estimation of $H_{\mathrm{R},\alpha}(\mathbf{y})$ ($\mathbf{y} \in \mathbb{R}^d$) using (i) k-nearest neighbors ($S = \{1, \ldots, k\}$; '`Renyi_kNN_1tok`') [66], (ii) generalized nearest neighbor graphs ($S \subseteq \{1, \ldots, k\}$; '`Renyi_kNN_S`') [64], (iii) weighted k-nearest neighbors ('`Renyi_weightedkNN`') [83], (iv) minimum spanning trees ('`Renyi_MST`') [115], and (v) geodesic spanning forests ('`Renyi_GSF`') [12]. The Tsallis entropy of a d-dimensional random variable $\mathbf{y}$ ($H_{\mathrm{T},\alpha}(\mathbf{y})$) can be estimated in ITE using the k-nearest neighbors method ($S = \{k\}$; '`Tsallis_kNN_k`') [45]. The multivariate Edgeworth expansion- [31] and the Voronoi region [50] based Shannon entropy estimators are also available in ITE ('`Shannon_Edgeworth`', '`Shannon_Voronoi`'). For the one-dimensional case ($d = 1$), beside the previous techniques, ITE offers sample spacing based estimators:

- Shannon entropy: by approximating the slope of the inverse distribution function [108] ('`Shannon_spacing_V`') and its bias corrected variant [18] ('`Shannon_spacing_Vb`'). The method described in [11] applies locally linear regression ('`Shannon_spacing_LL`'). Piecewise constant/linear correction has been applied in [55] ('`Shannon_spacing_Vpconst`')/[16] ('`Shannon_spacing_Vplin`').
- Rényi entropy: The idea of [108] and the empiric entropy estimator of order $m$ has been recently generalized to Rényi entropies [111] ('`Renyi_spacing_V`', '`Renyi_spacing_E`'). A continuously differentiable sample spacing (CDSS) based quadratic Rényi entropy estimator was presented in [56] ('`qRenyi_CDSS`').

The base entropy estimators of the ITE package are summarized in Table 2; the calling syntax of these methods is the same as in Example 2 and Example 3, one only has to change '`Shannon_kNN_k`' (see Example 2) and '`Renyi_kNN_k`' (see Example 3) to the `cost_name` given in the last column of the table.

Note: the `Renyi_kNN_1tok`, `Renyi_kNN_S`, `Renyi_MST`, `Renyi_GSF` methods (see Table 2) estimate the $H_\alpha$ Rényi entropy up to an additive constant which depends on the dimension $d$ and $\alpha$, but *not* on the distribution. In certain cases, such additive constants can also be relevant. They can be approximated via Monte-Carlo simulations, the computations are available in ITE. Let us take the example of `Renyi_kNN_1tok`, the estimation instructions are as follows:

1. Set `co.alpha` ($\alpha$) and `co.k` ($k$) in '`HRenyi_kNN_1tok_initialization.m`'.
2. Estimate the additive constant $\beta = \beta(d, k, \alpha)$ using '`estimate_HRenyi_constant.m`'.
3. Set the relevance of additive constants in the initialization function '`HRenyi_kNN_1tok_initialization.m`': '`co.additive_constant_is_relevant = 1`'.
4. Estimate the Rényi entropy (after initialization): '`HRenyi_kNN_1tok_estimation.m`'.

| Estimated quantity | Principle | $d$ | `cost_name` |
|---|---|---|---|
| Shannon entropy ($H$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'Shannon_kNN_k'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'Renyi_kNN_k'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | k-nearest neighbors ($S = \{1, \ldots, k\}$) | $d \geq 1$ | `'Renyi_kNN_1tok'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | generalized nearest neighbor graphs ($S \subseteq \{1, \ldots, k\}$) | $d \geq 1$ | `'Renyi_kNN_S'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | weighted k-nearest neighbors | $d \geq 1$ | `'Renyi_weightedkNN'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | minimum spanning trees | $d \geq 1$ | `'Renyi_MST'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | geodesic spanning forests | $d \geq 1$ | `'Renyi_GSF'` |
| Tsallis entropy ($H_{\mathrm{T},\alpha}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'Tsallis_kNN_k'` |
| Shannon entropy ($H$) | multivariate Edgeworth expansion | $d \geq 1$ | `'Shannon_Edgeworth'` |
| Shannon entropy ($H$) | Voronoi regions | $d \geq 2$ | `'Shannon_Voronoi'` |
| Shannon entropy ($H$) | approximate slope of the inverse distribution function | $d = 1$ | `'Shannon_spacing_V'` |
| Shannon entropy ($H$) | a bias corrected version of `'Shannon_spacing_V'` | $d = 1$ | `'Shannon_spacing_Vb'` |
| Shannon entropy ($H$) | `'Shannon_spacing_V'` with piecewise constant correction | $d = 1$ | `'Shannon_spacing_Vpconst'` |
| Shannon entropy ($H$) | `'Shannon_spacing_V'` with piecewise linear correction | $d = 1$ | `'Shannon_spacing_Vplin'` |
| Shannon entropy ($H$) | locally linear regression | $d = 1$ | `'Shannon_spacing_LL'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | extension of `'Shannon_spacing_V'` to $H_{\mathrm{R},\alpha}$ | $d = 1$ | `'Renyi_spacing_V'` |
| Rényi entropy ($H_{\mathrm{R},\alpha}$) | empiric entropy estimator of order $m$ | $d = 1$ | `'Renyi_spacing_E'` |
| quadratic Rényi entropy ($H_{\mathrm{R},2}$) | continuously differentiable sample spacing | $d = 1$ | `'qRenyi_CDSS'` |

Table 2: Entropy estimators (base). Third column: dimension ($d$) constraint.

### 3.1.2 Mutual Information Estimators

In our next example, we consider the estimation of the mutual information of the $d_m$-dimensional components of the random variable $\mathbf{y} = \left[\mathbf{y}^1, \ldots, \mathbf{y}^M\right] \in \mathbb{R}^d$ ($d = \sum_{m=1}^M d_m$):

$$I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = \int_{\mathbb{R}^{d_1}} \cdots \int_{\mathbb{R}^{d_M}} f\left(\mathbf{u}^1, \ldots, \mathbf{u}^M\right) \log \left[\frac{f\left(\mathbf{u}^1, \ldots, \mathbf{u}^M\right)}{\prod_{m=1}^M f_m(\mathbf{u}^m)}\right] \mathrm{d}\mathbf{u}^1 \cdots \mathrm{d}\mathbf{u}^M \tag{7}$$

using an i.i.d. sample set $\{\mathbf{y}_t\}_{t=1}^T$ from $\mathbf{y}$, where $f$ is the joint density function of $\mathbf{y}$ and $f_m$ is its $m^{th}$ marginal density, the density function of $\mathbf{y}^m$. As it is known, $I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right)$ is non-negative and is zero, if and only if the $\{\mathbf{y}^m\}_{m=1}^M$ variables are jointly independent [14]. Mutual information can be efficiently estimated, e.g., on the basis of entropy [Eq. (1)] or Kullback-Leibler divergence; we will return to these *derived* approaches while presenting *meta* estimators in Section 3.2.

There also exist other mutual information-like quantities measuring the independence of $\mathbf{y}^m$s:

1. **Kernel canonical correlation analysis (KCCA)**: The KCCA measure is defined as

$$I_{\mathrm{KCCA}}(\mathbf{y}^1, \mathbf{y}^2) = \sup_{f_1 \in \mathcal{F}^1, f_2 \in \mathcal{F}^2} \frac{\mathrm{cov}[f_1(\mathbf{y}^1), f_2(\mathbf{y}^2)]}{\sqrt{\mathrm{var}\left[f_1(\mathbf{y}^1)\right] + \kappa \left\|f_1\right\|_{\mathcal{F}^1}^2} \sqrt{\mathrm{var}\left[f_2(\mathbf{y}^2)\right] + \kappa \left\|f_2\right\|_{\mathcal{F}^2}^2}}, \quad (\kappa > 0) \tag{8}$$

   for $M = 2$ components, where 'cov' denotes covariance and 'var' stands for variance. In words, $I_{\mathrm{KCCA}}$ is the regularized form of the supremum correlation of $\mathbf{y}^1 \in \mathbb{R}^{d_1}$ and $\mathbf{y}^2 \in \mathbb{R}^{d_2}$ over two 'rich enough' reproducing kernel Hilbert spaces (RKHSs), $\mathcal{F}^1$ and $\mathcal{F}^2$. The computation of $I_{\mathrm{KCCA}}$ can be reduced to a generalized eigenvalue problem and the measure can be extended to $M \geq 2$ components to measure pairwise independence [4, 93]. The cost is called `'KCCA'` in ITE.

2. **Kernel generalized variance (KGV)**: Let $\mathbf{y} = \left[\mathbf{y}^1; \ldots, \mathbf{y}^M\right]$ be a multidimensional Gaussian random variable with covariance matrix $\mathbf{C}$ and let $\mathbf{C}^{i,j} \in \mathbb{R}^{d_i \times d_j}$ denote the cross-covariance between components of $\mathbf{y}^m \in \mathbb{R}^{d_m}$. In the Gaussian case, the mutual information between components $\mathbf{y}^1, \ldots, \mathbf{y}^M$ is [14]:

$$I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = -\frac{1}{2} \log \left(\frac{\det \mathbf{C}}{\prod_{m=1}^M \det \mathbf{C}^{m,m}}\right). \tag{9}$$

   If $\mathbf{y}$ is *not normal* then one can transform $\mathbf{y}^m$s using feature mapping $\boldsymbol{\varphi}$ associated with an RKHS and apply

Gaussian approximation to obtain

$$I_{\mathrm{KGV}}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = -\frac{1}{2}\log\left[\frac{\det(\boldsymbol{\mathcal{K}})}{\prod_{m=1}^{M}\det(\boldsymbol{\mathcal{K}}^{m,m})}\right], \tag{10}$$

where $\boldsymbol{\phi}(\mathbf{y}) := [\boldsymbol{\varphi}(\mathbf{y}^1);\ldots;\boldsymbol{\varphi}(\mathbf{y}^M)]$, $\boldsymbol{\mathcal{K}} := cov[\boldsymbol{\phi}(\mathbf{y})]$, and the sub-matrices are $\boldsymbol{\mathcal{K}}^{i,j} = cov[\boldsymbol{\varphi}(\mathbf{y}^i),\boldsymbol{\varphi}(\mathbf{y}^j)]$. For further details on the KGV method, see [4, 93]. The objective is called '`KGV`' in ITE.

3. **Hilbert-Schmidt independence criterion (HSIC)**: Let us given two separable RKHSs $\mathcal{F}^1$ and $\mathcal{F}^2$ with associated feature maps $\boldsymbol{\varphi}_1$ and $\boldsymbol{\varphi}_2$. Let the corresponding cross-covariance operator be

$$\mathbf{C}_{\mathbf{y}^1,\mathbf{y}^2} = \mathbb{E}\left(\left[\boldsymbol{\varphi}_1(\mathbf{y}^1)-\boldsymbol{\mu}_1\right]\otimes\left[\boldsymbol{\varphi}_2(\mathbf{y}^2)-\boldsymbol{\mu}_2\right]\right), \tag{11}$$

where $\otimes$ denotes tensor product, $\mathbb{E}$ is the expectation and the mean embeddings are

$$\boldsymbol{\mu}_m = \mathbb{E}[\boldsymbol{\varphi}_m(\mathbf{y}^m)] \quad (m=1,2). \tag{12}$$

HSIC [25] is defined as the Hilbert-Schmidt norm of the cross-covariance operator

$$I_{\mathrm{HSIC}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \left\|\mathbf{C}_{\mathbf{y}^1,\mathbf{y}^2}\right\|_{\mathrm{HS}}^2. \tag{13}$$

The HSIC measure can also be extended to the $M\geq 2$ case to measure pairwise independence; the objective is called '`HSIC`' in ITE.

Note: one can express HSIC in terms of pairwise similarities as

$$\begin{aligned}\left[I_{\mathrm{HSIC}}\left(\mathbf{y}^1,\mathbf{y}^2\right)\right]^2 &= \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{1'},\mathbf{y}^{2'}}k_1\left(\mathbf{y}^1,\mathbf{y}^{1'}\right)k_2\left(\mathbf{y}^2,\mathbf{y}^{2'}\right)+\mathbb{E}_{\mathbf{y}^1}\mathbb{E}_{\mathbf{y}^{1'}}k_1\left(\mathbf{y}^1,\mathbf{y}^{1'}\right)\mathbb{E}_{\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{2'}}k_1\left(\mathbf{y}^2,\mathbf{y}^{2'}\right)\\ &\quad -2\mathbb{E}_{\mathbf{y}^{1'}\mathbf{y}^{2'}}\left[\mathbb{E}_{\mathbf{y}^1}k_1\left(\mathbf{y}^1,\mathbf{y}^{1'}\right)\mathbb{E}_{\mathbf{y}^1}k_2\left(\mathbf{y}^2,\mathbf{y}^{2'}\right)\right],\end{aligned} \tag{14}$$

where (i) $k_i$-s are the reproducing kernels corresponding to $\mathcal{F}_i$-s, (ii) $\mathbf{y}^{i'}$ is an identical copy (in distribution) of $\mathbf{y}^i$ $(i=1,2)$.

4. **Generalized variance (GV)**: The GV measure [98] considers the decorrelation of two one-dimensional random variables $y^1\in\mathbb{R}$ and $y^2\in\mathbb{R}$ $(M=2)$ over a finite function set $\mathcal{F}$:

$$I_{\mathrm{GV}}\left(y^1,y^2\right) = \sum_{f\in\mathcal{F}}\left(corr\left[f\left(y^1\right),f\left(y^2\right)\right]\right)^2. \tag{15}$$

The name of the cost is '`GV`' in ITE.

5. **Hoeffding's $\Phi$, Schweizer-Wolff's $\sigma$ and $\kappa$**: Let $C$ be the copula of the random variable $\mathbf{y}=\left[y^1;\ldots;y^d\right]\in\mathbb{R}^d$. One may think of $C$ as the distribution function on $[0,1]^d$, which links the joint distribution function $(F)$ and the marginals $(F_i,\ i=1,\ldots,d)$:

$$F(\mathbf{y}) = C\left(F_1\left(y^1\right),\ldots,F_d\left(y^d\right)\right). \tag{16}$$

It can be shown that the $y^i\in\mathbb{R}$ variables are independent if and only if $C$, the copula of $\mathbf{y}$ equals to the product copula $\Pi$ defined as

$$\Pi(u_1,\ldots,u_d) = \prod_{i=1}^{d}u_i. \tag{17}$$

Using this result, the independence of $y^i$s can be measured by the (normalized) $L^p$ distance of $C$ and $\Pi$:

$$\left(h_p(d)\int_{[0,1]^d}|C(\mathbf{u})-\Pi(\mathbf{u})|^p\,\mathrm{d}\mathbf{u}\right)^{\frac{1}{p}}, \tag{18}$$

where (i) $1\leq p\leq\infty$, (ii) by an appropriate choice of the normalization constant $h_p(d)$, the value of (18) belongs to $\in[0,1]$ for any $C$.

- For $p = 2$, the special

$$I_\Phi\left(y^1,\ldots,y^d\right) = I_\Phi(C) = \left(h_2(d)\int_{[0,1]^d}[C(\mathbf{u})-\Pi(\mathbf{u})]^2\mathrm{d}\mathbf{u}\right)^{\frac{1}{2}} \tag{19}$$

quantity
  - is a generalization of Hoeffding's $\Phi$ defined for $d = 2$ [29],
  - can be analytically computed [22].

  The name of the objective is '`Hoeffding`' in ITE.
- For $p = 1$ and $p = \infty$, we obtain the Schweizer-Wolff's $\sigma$ and $\kappa$ [76], respectively. In this case no explicit expressions for the integrals are available. For small dimensional problems, however, the quantities can be efficiently estimated numerically. ITE contains methods for the $M = 2$ case:

$$I_{\mathrm{SW1}}\left(y^1,y^2\right) = I_{\mathrm{SW1}}(C) = \sigma = 12\int_{[0,1]^2}|C(\mathbf{u})-\Pi(\mathbf{u})|\mathrm{d}\mathbf{u}, \tag{20}$$

$$I_{\mathrm{SWinf}}\left(y^1,y^2\right) = I_{\mathrm{SWinf}}(C) = \kappa = 4\sup_{\mathbf{u}\in[0,1]^2}|C(\mathbf{u})-\Pi(\mathbf{u})|. \tag{21}$$

  In ITE the measures are available as '`SW1`' and '`SWinf`', respectively.

For an excellent introduction on copulas, see [52].

6. **Cauchy-Schwartz quadratic mutual information (QMI), Euclidean distance based QMI**: These measures are defined for the $\mathbf{y}^m \in \mathbb{R}^{d_m}$ $(m = 1, 2)$ variables as [78]:

$$I_{\text{QMI-CS}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \log\left[\frac{\left(\int_{\mathbb{R}^{d_1}}\int_{\mathbb{R}^{d_2}}\left[f\left(\mathbf{u}^1,\mathbf{u}^2\right)\right]^2\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{u}^2\right)\left(\int_{\mathbb{R}^{d_1}}\int_{\mathbb{R}^{d_2}}\left[f_1\left(\mathbf{u}^1\right)\right]^2\left[f_2\left(\mathbf{u}^2\right)\right]^2\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{u}^2\right)}{\left[\int_{\mathbb{R}^{d_1}}\int_{\mathbb{R}^{d_2}}f\left(\mathbf{u}^1,\mathbf{u}^2\right)f_1\left(\mathbf{u}^1\right)f_2\left(\mathbf{u}^2\right)\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{u}^2\right]^2}\right], \tag{22}$$

$$I_{\text{QMI-ED}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \left(\int_{\mathbb{R}^{d_1}}\int_{\mathbb{R}^{d_2}}\left[f\left(\mathbf{u}^1,\mathbf{u}^2\right)\right]^2\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{u}^2\right) + \left(\int_{\mathbb{R}^{d_1}}\int_{\mathbb{R}^{d_2}}\left[f_1\left(\mathbf{u}^1\right)\right]^2\left[f_2\left(\mathbf{u}^2\right)\right]^2\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{u}^2\right) \tag{23}$$

$$-2\int_{\mathbb{R}^{d_1}}\int_{\mathbb{R}^{d_2}}f\left(\mathbf{u}^1,\mathbf{u}^2\right)f_1\left(\mathbf{u}^1\right)f_2\left(\mathbf{u}^2\right)\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{u}^2. \tag{24}$$

   The measures can

   (a) be approximated in ITE via

$$\hat{f}_m(\mathbf{u}) = \frac{1}{T}\sum_{t=1}^{T}k\left(\mathbf{u}-\mathbf{y}_t^m\right) \tag{25}$$

   KDE (kernel density estimation; also termed the Parzen or the Parzen-Rosenblatt window method) in a plug-in scheme, directly or applying incomplete Cholesky decomposition ('`QMI_CS_KDE_direct`', '`QMI_CS_KDE_iChol`', '`QMI_ED_KDE_iChol`').

   (b) also be expressed in terms of the Cauchy-Schwartz and the Euclidean distance based divergences [see Eq. (48), (49)]:

$$I_{\text{QMI-CS}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = D_{\mathrm{CS}}\left(f, f_1f_2\right), \tag{26}$$

$$I_{\text{QMI-ED}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = D_{\mathrm{ED}}\left(f, f_1f_2\right). \tag{27}$$

7. **Distance covariance, distance correlation**: Two random variables are independent, if and only if their joint characteristic function can be factorized. This is the guiding principle behind the definition of distance covariance and distance correlation [102, 99]. Namely, let us given $\mathbf{y}^1 \in \mathbb{R}^{d_1}$, $\mathbf{y}^2 \in \mathbb{R}^{d_2}$ random variables $(M = 2)$, and let $\varphi_j$ $(\varphi_{12})$ stand for the characteristic function of $\mathbf{y}^j$ $(\left[\mathbf{y}^1;\mathbf{y}^2\right])$:

$$\varphi_{12}\left(\mathbf{u}^1,\mathbf{u}^2\right) = \mathbb{E}\left[e^{i\langle\mathbf{u}^1,\mathbf{y}^1\rangle+i\langle\mathbf{u}^2,\mathbf{y}^2\rangle}\right], \tag{28}$$

$$\varphi_j\left(\mathbf{u}^j\right) = \mathbb{E}\left[e^{i\langle\mathbf{u}^j,\mathbf{y}^j\rangle}\right], \quad (j = 1, 2) \tag{29}$$

$$\tag{30}$$

where $i = \sqrt{-1}$, $\langle \cdot, \cdot \rangle$ is the standard Euclidean scalar product, and $\mathbb{E}$ stands for expectation. The *distance covariance* is simply the $L^2(w)$ norm of $\varphi_{12}$ and $\varphi_1\varphi_2$:

$$I_{\mathrm{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \left\|\varphi_{12} - \varphi_1\varphi_2\right\|_{L^2(w)} = \sqrt{\int_{\mathbb{R}^{d_1+d_2}} \left|\varphi_{12}\left(\mathbf{u}^1, \mathbf{u}^2\right) - \varphi_1\left(\mathbf{u}^1\right)\varphi_2\left(\mathbf{u}^2\right)\right|^2 w\left(\mathbf{u}^1, \mathbf{u}^2\right) \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2} \tag{31}$$

with a suitable chosen $w$ weight function

$$w\left(\mathbf{u}^1, \mathbf{u}^2\right) = \frac{1}{c(d_1,\alpha)c(d_2,\alpha)\left[\left\|\mathbf{u}^1\right\|_2\right]^{d_1+\alpha}\left[\left\|\mathbf{u}^2\right\|_2\right]^{d_2+\alpha}}, \tag{32}$$

where $\alpha \in (0,2)$ and

$$c(d,\alpha) = \frac{2\pi^{\frac{d}{2}}\Gamma\left(1-\frac{\alpha}{2}\right)}{\alpha 2^{\alpha}\Gamma\left(\frac{d+\alpha}{2}\right)}. \tag{33}$$

The *distance variance* is defined analogously ($j = 1, 2$):

$$I_{\mathrm{dVar}}\left(\mathbf{y}^j, \mathbf{y}^j\right) = \left\|\varphi_{jj} - \varphi_j\varphi_j\right\|_{L^2(w)}. \tag{34}$$

The *distance correlation* is the standardized version of the distance covariance:

$$I_{\mathrm{dCor}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \begin{cases} \frac{I_{\mathrm{dCov}}\left(\mathbf{y}^1,\mathbf{y}^2\right)}{\sqrt{I_{\mathrm{dVar}}\left(\mathbf{y}^1,\mathbf{y}^1\right) I_{\mathrm{dVar}}\left(\mathbf{y}^2,\mathbf{y}^2\right)}}, & \text{if } I_{\mathrm{dVar}}\left(\mathbf{y}^1,\mathbf{y}^1\right) I_{\mathrm{dVar}}\left(\mathbf{y}^2,\mathbf{y}^2\right) > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{35}$$

a type of unsigned correlation. By construction $I_{\mathrm{dCor}}\left(\mathbf{y}^1, \mathbf{y}^2\right) \in [0,1]$, and is zero, if and only if $\mathbf{y}^1$ and $\mathbf{y}^2$ are independent. The distance covariance and distance correlation measures are called '`dCov`' and '`dCor`' in ITE.

The estimation of these quantities can be carried out easily in the ITE package. Let us take the KCCA measure as an example:

**Example 4 (Mutual information estimation (base: usage))**

```
>ds = [2;3;4]; Y = rand(sum(ds),5000); %generate the data of interest (ds(m)=dim(y^m), T=5000)
>mult = 1;                             %multiplicative constant is important
>co = IKCCA_initialization(mult);      %initialize the mutual information ('I') estimator ('KCCA')
>I = IKCCA_estimation(Y,ds,co);        %perform mutual information estimation
```

The calling syntax of the mutual information estimators, are completely the same; one only has to change '`KCCA`' to the `cost_name` given in the last column of the Table 3. The table summarizes the base mutual information estimators in ITE.

### 3.1.3 Divergence Estimators

Divergences measure the 'distance' between two probability densities, $f_1 : \mathbb{R}^d \mapsto \mathbb{R}$ and $f_2 : \mathbb{R}^d \mapsto \mathbb{R}$. One of the most well-known such index is the Kullback-Leibler divergence (also called relative entropy) [42]:

$$D(f_1, f_2) = \int_{\mathbb{R}^d} f_1(\mathbf{u}) \log\left[\frac{f_1(\mathbf{u})}{f_2(\mathbf{u})}\right] \mathrm{d}\mathbf{u}. \tag{36}$$

In practise, one has independent, i.i.d. samples from $f_1$ and $f_2$, $\{\mathbf{y}_t^1\}_{t=1}^{T_1}$ and $\{\mathbf{y}_t^2\}_{t=1}^{T_2}$, respectively. The goal is to estimate divergence $D$ using these samples. Of course, there exist many variants/extensions of the traditional Kullback-Leibler divergence [109, 5]; depending on the application addressed, different divergences can be advantageous. The ITE package is capable of estimating the following divergences, too:

1. $L_2$ **divergence**:

$$D_{\mathrm{L}}(f_1, f_2) = \sqrt{\int_{\mathbb{R}^d} \left[f_1(\mathbf{u}) - f_2(\mathbf{u})\right]^2 \mathrm{d}\mathbf{u}}. \tag{37}$$

| Estimated quantity | Principle | $d_m$ | $M$ | `cost_name` |
|---|---|---|---|---|
| generalized variance ($I_{\mathrm{GV}}$) | f-covariance/-correlation ($f \in \mathcal{F}$, $\lvert\mathcal{F}\rvert < \infty$) | $d_m = 1$ | $M = 2$ | `'GV'` |
| Hilbert-Schmidt indep. criterion ($I_{\mathrm{HSIC}}$) | HS norm of the cross-covariance operator | $d_m \geq 1$ | $M \geq 2$ | `'HSIC'` |
| kernel canonical correlation ($I_{\mathrm{KCCA}}$) | sup correlation over RKHSs | $d_m \geq 1$ | $M \geq 2$ | `'KCCA'` |
| kernel generalized variance ($I_{\mathrm{KGV}}$) | Gaussian mutual information of the features | $d_m \geq 1$ | $M \geq 2$ | `'KGV'` |
| Hoeffding's $\Phi$ ($I_{\Phi}$), multivariate | $L^2$ distance of the joint- and the product copula | $d_m = 1$ | $M \geq 2$ | `'Hoeffding'` |
| Schweizer-Wolff's $\sigma$ ($I_{\mathrm{SW1}}$) | $L^1$ distance of the joint- and the product copula | $d_m = 1$ | $M = 2$ | `'SW1'` |
| Schweizer-Wolff's $\kappa$ ($I_{\mathrm{SWinf}}$) | $L^\infty$ distance of the joint- and the product copula | $d_m = 1$ | $M = 2$ | `'SWinf'` |
| Cauchy-Schwartz QMI ($I_{\mathrm{QMI\text{-}CS}}$) | KDE, direct | $d_m = 1$ | $M = 2$ | `'QMI_CS_KDE_direct'` |
| Cauchy-Schwartz QMI ($I_{\mathrm{QMI\text{-}CS}}$) | KDE, incomplete Cholesky decomposition | $d_m \geq 1$ | $M = 2$ | `'QMI_CS_KDE_iChol'` |
| Euclidean dist. based QMI ($I_{\mathrm{QMI\text{-}ED}}$) | KDE, incomplete Cholesky decomposition | $d_m \geq 1$ | $M = 2$ | `'QMI_ED_KDE_iChol'` |
| distance covariance ($I_{\mathrm{dCov}}$) | pairwise distances | $d_m \geq 1$ | $M = 2$ | `'dCov'` |
| distance correlation ($I_{\mathrm{dCor}}$) | pairwise distances | $d_m \geq 1$ | $M = 2$ | `'dCor'` |

Table 3: Mutual information estimators (base). Third column: dimension constraint ($d_m$; $\mathbf{y}^m \in \mathbb{R}^{d_m}$). Fourth column: constraint for the number of components ($M$; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$).

2. **Tsallis divergence**:

$$D_{\mathrm{T},\alpha}(f_1, f_2) = \frac{1}{\alpha - 1}\left(\int_{\mathbb{R}^d} f_1^{\alpha}(\mathbf{u}) f_2^{1-\alpha}(\mathbf{u}) \mathrm{d}\mathbf{u} - 1\right) \quad (\alpha \in \mathbb{R} \setminus \{1\}). \tag{38}$$

The Kullback-Leibler divergence [Eq. (36)] is a special of Tsallis' in limit sense:

$$\lim_{\alpha \to 1} D_{\mathrm{T},\alpha} = D. \tag{39}$$

3. **Rényi divergence**:

$$D_{\mathrm{R},\alpha}(f_1, f_2) = \frac{1}{\alpha - 1} \log \int_{\mathbb{R}^d} f_1^{\alpha}(\mathbf{u}) f_2^{1-\alpha}(\mathbf{u}) \mathrm{d}\mathbf{u} \quad (\alpha \in \mathbb{R} \setminus \{1\}). \tag{40}$$

The Kullback-Leibler divergence [Eq. (36)] is a special of Rényi's in limit sense:

$$\lim_{\alpha \to 1} D_{\mathrm{R},\alpha} = D. \tag{41}$$

4. **Maximum mean discrepancy** (MMD, also called the kernel distance) [24]:

$$D_{\mathrm{MMD}}(f_1, f_2) = \left\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right\|_{\mathcal{F}}, \tag{42}$$

where $\boldsymbol{\mu}_m$ is the mean embedding of $f_m$ ($m = 1, 2$) and $\mathcal{F} = \mathcal{F}^1 = \mathcal{F}^2$, see the definition of HSIC [Eq. (12)]. Notes:

- In the statistics literature, MMD is known as an integral probability metric (IPM) [117, 51, 84]:

$$D_{\mathrm{MMD}}(f_1, f_2) = \sup_{g \in \mathcal{B}} \left(\mathbb{E}\left[g\left(\mathbf{y}^1\right)\right] - \mathbb{E}[g\left(\mathbf{y}^2\right)]\right), \tag{43}$$

  where $f_i$ is the density of $\mathbf{y}^i$ ($i = 1, 2$) and $\mathcal{B}$ is the unit ball in the RKHS $\mathcal{F}$.
- One can easily see that the MMD measure acts as a 'divergence' on the joint and the product of the marginals in HSIC (similarly to the well-known Kullback-Leibler divergence and its extensions, see Eqs. (77)-(78)):

$$I_{\mathrm{HSIC}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = D_{\mathrm{MMD}}(f, f_1 f_2), \tag{44}$$

  where $f$ is the joint density of $\left[\mathbf{y}^1; \mathbf{y}^2\right]$.
- In terms of pairwise similarities MMD satisfies the relation:

$$\left[D_{\mathrm{MMD}}(f_1, f_2)\right]^2 = \mathbb{E}_{\mathbf{y}^1, \mathbf{y}^{1'}}\left[k\left(\mathbf{y}^1, \mathbf{y}^{1'}\right)\right] + \mathbb{E}_{\mathbf{y}^2, \mathbf{y}^{2'}}\left[k\left(\mathbf{y}^2, \mathbf{y}^{2'}\right)\right] - 2\mathbb{E}_{\mathbf{y}^1, \mathbf{y}^2}\left[k\left(\mathbf{y}^1, \mathbf{y}^2\right)\right], \tag{45}$$

  where $\mathbf{y}^{i'}$ is an identical copy (in distribution) of $\mathbf{y}^i$ ($i = 1, 2$).

5. **Hellinger distance**:

$$D_{\mathrm{H}}(f_1,f_2)=\sqrt{\frac{1}{2}\int_{\mathbb{R}^d}\left[\sqrt{f_1(\mathbf{u})}-\sqrt{f_2(\mathbf{u})}\right]^2\mathrm{d}\mathbf{u}}=\sqrt{1-\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}}. \tag{46}$$

6. **Bhattacharyya distance**:

$$D_{\mathrm{B}}(f_1,f_2)=-\log\left(\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}\right). \tag{47}$$

7. **Cauchy-Schwartz and Euclidean distance based divergences**:

$$D_{\mathrm{CS}}(f_1,f_2)=\log\left[\frac{\left(\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^2\mathrm{d}\mathbf{u}\right)\left(\int_{\mathbb{R}^d}\left[f_2(\mathbf{u})\right]^2\mathrm{d}\mathbf{u}\right)}{\left(\int_{\mathbb{R}^d}f_1(\mathbf{u})f_2(\mathbf{u})\mathrm{d}\mathbf{u}\right)^2}\right]=\log\left[\frac{1}{\cos^2(f_1,f_2)}\right], \tag{48}$$

$$D_{\mathrm{ED}}(f_1,f_2)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^2\mathrm{d}\mathbf{u}+\int_{\mathbb{R}^d}\left[f_2(\mathbf{u})\right]^2\mathrm{d}\mathbf{u}-2\int_{\mathbb{R}^d}f_1(\mathbf{u})f_2(\mathbf{u})\mathrm{d}\mathbf{u}=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})-f_2(\mathbf{u})\right]^2\mathrm{d}\mathbf{u} \tag{49}$$

$$=[D_{\mathrm{L}}(f_1,f_2)]^2. \tag{50}$$

8. **Energy distance**: Let $(\mathcal{Z},\rho)$ be a semimetric space of negative type (see Def. 2, Section D), and let $\mathbf{y}^1$ and $\mathbf{y}^2$ be $\mathcal{Z}$-valued random variables with (i) densities $f_1$ and $f_2$, and (ii) let $\mathbf{y}^{1'}$ and $\mathbf{y}^{2'}$ be an identically distributed copy of $\mathbf{y}^1$ and $\mathbf{y}^2$, respectively. The energy distance of $\mathbf{y}^1$ and $\mathbf{y}^2$ is defined as [100, 101]:

$$D_{\mathrm{EnDist}}(f_1,f_2)=2\mathbb{E}\left[\rho\left(\mathbf{y}^1,\mathbf{y}^2\right)\right]-\mathbb{E}\left[\rho\left(\mathbf{y}^1,\mathbf{y}^{1'}\right)\right]-\mathbb{E}\left[\rho\left(\mathbf{y}^2,\mathbf{y}^{2'}\right)\right]. \tag{51}$$

An important special case is the Euclidean ($\mathcal{Z}=\mathbb{R}^d$ with $\|\cdot\|_2$), when the energy distance takes the form:

$$D_{\mathrm{EnDist}}(f_1,f_2)=2\mathbb{E}\left\|\mathbf{y}^1-\mathbf{y}^2\right\|_2-\mathbb{E}\left\|\mathbf{y}^1-\mathbf{y}^{1'}\right\|_2-\mathbb{E}\left\|\mathbf{y}^2-\mathbf{y}^{2'}\right\|_2. \tag{52}$$

In the further specialized $d=1$ case, the energy distance equals to twice the Cramer-Von Mises distance. The energy distance

- is non-negative; and in case of *strictly* negative space $\mathcal{Z}$ (e.g., $\mathbb{R}^d$) it is zero, if and only if $\mathbf{y}^1$ and $\mathbf{y}^2$ are identically distributed,
- in ITE it is called '`EnergyDist`'.

Let us note that for (38), (40), (46) and (47), it is sufficient to estimate the

$$D_{\mathrm{temp1}}(\alpha)=\int_{\mathbb{R}^d}[f_1(\mathbf{u})]^\alpha[f_2(\mathbf{u})]^{1-\alpha}\mathrm{d}\mathbf{u} \tag{53}$$

quantity, which is called the Bhattacharyya coefficient for $\alpha=\frac{1}{2}$ (it is also called the Hellinger affinity; see (46) and (47)):

$$BC=\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}\in[0,1]. \tag{54}$$

(53) can also be further generalized to

$$D_{\mathrm{temp2}}(a,b)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^a\left[f_2(\mathbf{u})\right]^b f_1(\mathbf{u})\mathrm{d}\mathbf{u},\quad(a,b\in\mathbb{R}). \tag{55}$$

The calling syntax of the divergence estimators in the ITE package are again uniform. In the following example, the estimation of the Rényi divergence is illustrated using the k-nearest neighbor method:

**Example 5 (Divergence estimation (base: usage))**

```
>Y1 = randn(3,2000); Y2 = randn(3,3000); %generate the data of interest (d=3, T_1=2000, T_2=3000)
>mult = 1;                               %multiplicative constant is important
>co = DRenyi_kNN_k_initialization(mult); %initialize the divergence ('D') estimator ('Renyi_kNN_k')
>D = DRenyi_kNN_k_estimation(Y1,Y2,co);  %perform divergence estimation
```

| Estimated quantity | Principle | $d$ | `cost_name` |
|---|---|---|---|
| $L_2$ divergence ($D_{\mathrm{L}}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'L2_kNN_k'` |
| Tsallis divergence ($D_{\mathrm{T},\alpha}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'Tsallis_kNN_k'` |
| Rényi divergence ($D_{\mathrm{R},\alpha}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'Renyi_kNN_k'` |
| maximum mean discrepancy ($D_{\mathrm{MMD}}$) | U-statistics, unbiased | $d \geq 1$ | `'MMD_Ustat'` |
| maximum mean discrepancy ($D_{\mathrm{MMD}}$) | V-statistics, biased | $d \geq 1$ | `'MMD_Vstat'` |
| maximum mean discrepancy ($D_{\mathrm{MMD}}$) | online | $d \geq 1$ | `'MMD_online'` |
| Hellinger distance ($D_{\mathrm{H}}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'Hellinger_kNN_k'` |
| Bhattacharyya distance ($D_{\mathrm{B}}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'Bhattacharyya_kNN_k'` |
| Kullback-Leibler divergence ($D$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'KL_kNN_k'` |
| Kullback-Leibler divergence ($D$) | k-nearest neighbors ($S_i = \{k_i(T_i)\}$) | $d \geq 1$ | `'KL_kNN_kiTi'` |
| Cauchy-Schwartz divergence ($D_{\mathrm{CS}}$) | KDE, incomplete Cholesky decomposition | $d \geq 1$ | `'CS_KDE_iChol'` |
| Euclidean distance based divergence ($D_{\mathrm{ED}}$) | KDE, incomplete Cholesky decomposition | $d \geq 1$ | `'ED_KDE_iChol'` |
| energy distance ($D_{\mathrm{EnDist}}$) | pairwise distances | $d \geq 1$ | `'EnergyDist'` |

Table 4: Divergence estimators (base). Third column: dimension ($d$) constraint.

Beyond the Rényi divergence $D_{\mathrm{R},\alpha}$ [68, 67, 69] (`'Renyi_kNN_k'`), the k-nearest neighbor technique can also be used to estimate the $L_2$- ($D_{\mathrm{L}}$) [68, 67, 69] (`'L2_kNN_k'`), the Tsallis ($D_{\mathrm{T},\alpha}$) divergence [68, 67] (`'Tsallis_kNN_k'`), and of course, specially to the Kullback-Leibler divergence ($D$) [45, 58, 112] (`'KL_kNN_k'`, `'KL_kNN_kiTi'`). A similar approach can be applied to the estimation of the (55) quantity [63], specially to the Hellinger- and the Bhattacharyya distance (`'Hellinger_kNN_k'`, `'Bhattacharyya_kNN_k'`). For the MMD measure [24], (i) an U-statistic based (`'MMD_Ustat'`), (ii) a V-statistic based (`'MMD_Vstat'`), and (iii) a linearly scaling, online method (`'MMD_online'`) have been implemented in ITE. The Cauchy-Schwartz and the Euclidean distance based divergences ($D_{\mathrm{CS}}$, $D_{\mathrm{ED}}$) can be estimated using KDE based plug-in methods, applying incomplete Cholesky decomposition (`'CS_KDE_iChol'`, `''ED_KDE_iChol'`). The energy distance ($D_{\mathrm{EnDist}}$) can be approximated using pairwise distances of sample points (`'EnergyDist'`). Table 4 contains the base divergence estimators of the ITE package. The estimations can be carried out by changing the name `'Renyi_kNN_k'` in Example 5 to the `cost_name` given in the last column of the table.

### 3.1.4 Association Estimators

There exist many exciting association quantities measuring certain dependency relations of random variables – in ITE we think of mutual information (Section 3.1.2) as a special case of association that (i) is non-negative, (ii) being zero, if its arguments are independent.

Our goal is to estimate the dependence/association of the $d_m$-dimensional components of the random variable $\mathbf{y} = \left[\mathbf{y}^1, \ldots, \mathbf{y}^M\right] \in \mathbb{R}^d$ ($d = \sum_{m=1}^M d_m$), from which we have i.i.d. samples $\{\mathbf{y}_t\}_{t=1}^T$. One of the most well-known example of associations is that of the Spearman's $\rho$ (also called the Spearman's rank correlation coefficient, or the grade correlation coefficient) [82]. For $d = 2$, it is defined as

$$I_\rho\left(y^1, y^2\right) = corr\left(F_1\left(y^1\right), F_2\left(y^2\right)\right), \tag{56}$$

where 'corr' stands for correlation and $F_i$ denotes the distribution function (cdf) of $y^i$. Spearman's $\rho$ is a special association, a *measure of concordance*: if large (small) values of $y^1$ tend to be associated with large (small) values of $y^2$, it is reflected in $I_\rho$. For a formal definition of measures of concordance, see Def. 1 (Section D).

Let us now define for $d_m = 1$ ($\forall m$) the comonotonicy copula (also called the Fréchet-Hoeffding upper bound[9]) as

$$M(\mathbf{u}) = \min_{i=1,\ldots,d} u_i. \tag{57}$$

It is known that $I_\rho$ can be interpreted as the normalized average difference of the copula of $\mathbf{y}$ ($C$) and the independence copula ($\Pi$) [see Eq. (17)]:

$$A_\rho\left(y^1, y^2\right) = A_\rho(C) = \frac{\int_{[0,1]^2} u_1 u_2 \mathrm{d}C(\mathbf{u}) - \left(\frac{1}{2}\right)^2}{\frac{1}{12}} = 12 \int_{[0,1]^2} C(\mathbf{u})\mathrm{d}\mathbf{u} - 3 = \frac{\int_{[0,1]^2} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^2} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^2} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^2} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}, \tag{58}$$

[9] The name originates from the fact that for any $C$ copula $W(\mathbf{u}) := \max(u_1 + \ldots + u_d - d + 1, 0) \leq C(\mathbf{u}) \leq M(\mathbf{u})$, $(\forall \mathbf{u} \in [0,1]^d)$; $W$ is called the Fréchet-Hoeffding lower bound.

| Estimated quantity | Principle | $d_m$ | $M$ | cost_name |
|---|---|---|---|---|
| Spearman's $\rho$: multivariate1 ($A_{\rho_1}$) | empirical copula, explicit formula | $d_m = 1$ | $M \geq 2$ | 'Spearman1' |
| Spearman's $\rho$: multivariate2 ($A_{\rho_2}$) | empirical copula, explicit formula | $d_m = 1$ | $M \geq 2$ | 'Spearman2' |
| Spearman's $\rho$: multivariate3 ($A_{\rho_3}$) | $\rho_3$ is the average of $\rho_1$ and $\rho_2$ | $d_m = 1$ | $M \geq 2$ | 'Spearman3' |

Table 5: Association estimators (base). Third column: dimension constraint ($d_m$; $\mathbf{y}^m \in \mathbb{R}^{d_m}$). Fourth column: constraint for the number of components ($M$; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$).

where the

$$\int_{[0,1]^2} M(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{1}{3}, \qquad \int_{[0,1]^2} \Pi(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{1}{4} \tag{59}$$

properties were exploited. The association measures included in ITE are the following:

1. One can extend [74] the Spearman's $\rho$ to the multivariate case using (58):

$$A_{\rho_1}\left(y^1, \ldots, y^d\right) = A_{\rho_1}(C) = h_\rho(d)\left[2^d \int_{[0,1]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - 1\right], \tag{60}$$

   where

$$h_\rho(d) = \frac{d+1}{2^d - (d+1)}. \tag{61}$$

   The name of the association measure is 'Spearman1' in ITE.

2. An other multivariate extension of Spearman's $\rho$ is [74] using (58) is

$$A_{\rho_2}\left(y^1, \ldots, y^d\right) = A_{\rho_2}(C) = h_\rho(d)\left[2^d \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}C(\mathbf{u}) - 1\right]. \tag{62}$$

   The association measure is called 'Spearman2' in ITE.

3. [52] further considers the average of $A_{\rho_1}$ and $A_{\rho_2}$:

$$A_{\rho_3}\left(y^1, \ldots, y^d\right) = A_{\rho_3}(C) = \frac{A_{\rho_1}\left(y^1, \ldots, y^d\right) + A_{\rho_2}\left(y^1, \ldots, y^d\right)}{2}. \tag{63}$$

   The name of this association measure is 'Spearman3' in ITE[10]. For the special case of $d = 2$, the defined extensions of Spearman's $\rho$ coincide:

$$A_\rho = A_{\rho_1} = A_{\rho_2} = A_{\rho_3}. \tag{64}$$

The calling syntax of the association estimators is uniform and very simple; as an example the $A_{\rho_1}$ measure is estimated:

**Example 6 (Association estimation (base: usage))**

```
>ds = ones(3,1); Y = rand(sum(ds),5000); %generate the data of interest (ds(m)=dim(y^m), T=5000)
>mult = 1;                               %multiplicative constant is important
>co = ASpearman1_initialization(mult);   %initialize the association ('A') estimator ('Spearman1')
>A = ASpearman1_estimation(Y,ds,co);     %perform association estimation
```

For the estimation of other association measures it is sufficient to change 'Spearman1' to the cost_name given in the last column of Table 5 summarizing the base association estimators.

[10] Although (63) would make it possibile to implement $A_{\rho_3}$ as a meta estimator (see Section 3.2.4), for computational reasons (to not compute the same rank statistics twice), it became a base method.

| Estimated quantity | Principle | $d$ | `cost_name` |
|---|---|---|---|
| cross-entropy ($C_{\mathrm{CE}}$) | k-nearest neighbors ($S = \{k\}$) | $d \geq 1$ | `'CE_kNN_k'` |

Table 6: Cross estimators (base). Third column: dimension ($d$) constraint.

### 3.1.5 Cross Estimators

'Cross'-type measures arise naturally in information theory – we think of divergences (see Section 3.1.3) in ITE as a special class of cross measures which (i) are non-negative, (ii) being zero, if and only if $f_1 = f_2$. Our goal is to estimate such cross quantities from independent, i.i.d. samples $\left\{\mathbf{y}_t^1\right\}_{t=1}^{T_1}$ and $\left\{\mathbf{y}_t^2\right\}_{t=1}^{T_2}$ distributed according to $f_1$ and $f_2$, respectively. One of the most well-known such quantity is *cross-entropy*. The cross-entropy of two probability densities, $f_1 : \mathbb{R}^d \mapsto \mathbb{R}$ and $f_2 : \mathbb{R}^d \mapsto \mathbb{R}$ is defined as:

$$C_{\mathrm{CE}}(f_1, f_2) = -\int_{\mathbb{R}^d} f_1(\mathbf{u}) \log\left[f_2(\mathbf{u})\right] \mathrm{d}\mathbf{u}. \tag{65}$$

One can estimate $C_{\mathrm{CE}}$ via the k-nearest neighbor ($S = \{k\}$) technique [45]; the method is available in ITE and is called '`CE_kNN_k`'. The calling syntax of the cross estimators is uniform, an example is given below:

**Example 7 (Cross estimation (base: usage))**

```
>Y1 = randn(3,2000); Y2 = randn(3,3000); %generate the data of interest (d=3, T1=2000, T2=3000)
>mult = 1;                               %multiplicative constant is important
>co = CCE_kNN_k_initialization(mult);    %initialize the cross ('C') estimator ('CE_kNN_k')
>C = CCE_kNN_k_estimation(Y1,Y2,co);     %perform cross-entropy estimation
```

The base cross estimators of ITE are summarized in Table 6.

## 3.2 Meta Estimators

Here, we present how one can easily derive in the ITE package new information theoretical estimators from existing ones on the basis of relations between entropy, mutual information, divergence, association and cross quantities. These *meta* estimators are included in ITE. The additional goal of this section is to provide examples for meta estimator construction so that users could simply create novel ones. In Section 3.2.1, Section 3.2.2, Section 3.2.3, Section 3.2.4 and Section 3.2.5, we focus on entropy, mutual information, divergence, association and cross estimators, respectively.

### 3.2.1 Entropy Estimators

Here, we present the idea of the meta construction in entropy estimation through examples:

1. The first example considers estimation via the ensemble approach. As it has been recently demonstrated the computational load of entropy estimation can be heavily decreased by (i) dividing the available samples into groups and then (ii) computing the averages of the group estimates [43]. Formally, let the samples be denoted by $\{\mathbf{y}_t\}_{t=1}^{T}$ ($\mathbf{y}_t \in \mathbb{R}^d$) and let us partition them into $N$ groups of size $g$ ($gN = T$), $\{1, \ldots, T\} = \cup_{n=1}^{N} I_n$ ($I_i \cap I_j = \emptyset$, $i \neq j$) and average the estimations based on the groups

$$H_{\mathrm{ensemble}}(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \hat{H}\left(\{\mathbf{y}_t\}_{t \in I_n}\right). \tag{66}$$

   As a prototype example for meta entropy estimation the implementation of the ensemble method [Eq. (66)] is provided below (see Example 8 and Example 9). In the example, the individual estimators in the ensemble are based on k-nearest neighbors ('`Shannon_kNN_k`'). However, the flexibility of the ITE package allows to change the $H$ estimator [r.h.s of (66)] to *any* other entropy technique (base/meta, see Table 2 and Table 7).

   **Example 8 (Entropy estimation (meta: initialization))**

```
function [co] = Hensemble_initialization(mult)
co.name = 'ensemble';             %name of the estimator: 'ensemble'
co.mul = mult;                    %set whether multiplicative constant is important
co.group_size = 500;              %group size (g=500)
co.member_name = 'Shannon_kNN_k'; %estimator used in the ensemble ('Shannon_kNN_k')
co.member_co = H_initialization(co.member_name,mult);%initialize the member in the ensemble,
                                                     %the value of 'mult' is passed
```

The estimation part is carried out in accordance with (66):

**Example 9 (Entropy estimation (meta: estimation))**

```
function [H] = Hensemble_estimation(Y,co)
g = co.group_size;                          %initialize group size (g)
num_of_samples = size(Y,2);                 %initialize number of samples (T)
num_of_groups = floor(num_of_samples/g);    %initialize number of groups (N)

H = 0;
for k = 1 : num_of_groups                   %compute the average over the ensemble
  H = H + H_estimation(Y(:,(k-1)*g+1:k*g),co.member_co); %add the estimation
                                                         %of the initialized member
end
H = H / num_of_groups;
```

The usage of the defined method follows the syntax of base entropy estimators (Example 2, Example 3):

**Example 10 (Entropy estimation (meta: usage))**

```
>Y = rand(5,1000);                    %generate the data of interest (d=5, T=1000)
>mult = 1;                            %multiplicative constant is important
>co = Hensemble_initialization(mult); %initialize the entropy ('H') estimator ('ensemble'),
>H = Hensemble_estimation(Y,co);      %perform entropy estimation
```

2. Since (i) entropy can be estimated consistently using pairwise distances of sample points[11], and (ii) random projection (RP) techniques realize approximate isometric embeddings [36, 20, 34, 1, 46, 3, 49], one can construct efficient estimation methods by the integration of the ensemble and the RP technique.

   Formally, the definition of the estimation is identical to that of the ensemble approach [Eq. (66)], except for random projections $\mathbf{R}_n \in \mathbb{R}^{d_{RP} \times d}$ $(n = 1, \ldots, N)$. The final estimation is

$$H_{\text{RPensemble}}(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \hat{H}\left(\{\mathbf{R}_n \mathbf{y}_t\}_{t \in I_n}\right). \tag{67}$$

   The approach shows exciting potentials with serious computational speed-ups in independent subspace analysis [89] and image registration [90]. The technique has been implemented in the ITE toolbox under the name '`RPensemble`' (see Table 7, `HRPensemble_initialization.m`, `HRPensemble_estimation.m`).

3. Information theoretical quantities can be defined over the complex domain via the Hilbert transformation [17]

$$\varphi_v : \mathbb{C}^d \ni \mathbf{v} \mapsto \mathbf{v} \otimes \begin{bmatrix} \Re(\cdot) \\ \Im(\cdot) \end{bmatrix} \in \mathbb{R}^{2d}, \tag{68}$$

   as the entropy of the mapped 2d-dimensional real variable

$$H_{\mathbb{C}}(\mathbf{y}) := H(\varphi_v(\mathbf{y})). \tag{69}$$

   Relation (69) can be transformed to a meta entropy estimator, the method is available under the name '`complex`' (see Table 7, `Hcomplex_initialization.m`, `Hcomplex_estimation.m`).

[11] The construction holds for other information theoretical quantities like mutual information and divergence.

| Estimated quantity | Principle | $d$ | cost_name |
|---|---|---|---|
| complex entropy ($H_{\mathbb{C}}$) | entropy of a real random vector variable | $d \geq 1$ | 'complex' |
| Shannon entropy ($H$) | average the entropy over an ensemble | $d \geq 1$ | 'ensemble' |
| Shannon entropy ($H$) | average the entropy over a random projected ensemble | $d \geq 1$ | 'RPensemble' |
| Tsallis entropy ($H_{\mathrm{T},\alpha}$) | function of the Rényi entropy | $d \geq 1$ | 'Tsallis_HRenyi' |
| Shannon entropy ($H$) | -KL divergence from the normal distribution | $d \geq 1$ | 'Shannon_DKL_N' |
| Shannon entropy ($H$) | -KL divergence from the uniform distribution | $d \geq 1$ | 'Shannon_DKL_U' |

Table 7: Entropy estimators (meta). Third column: dimension ($d$) constraint.

4. Using (3) and (5), the Tsallis entropy can be computed from the Rényi entropy:

$$H_{\mathrm{T},\alpha}(\mathbf{y}) = \frac{e^{(1-\alpha)H_{\mathrm{R},\alpha}(\mathbf{y})} - 1}{1-\alpha}. \tag{70}$$

This relation is realized in ITE by the 'Tsallis_HRenyi' meta entropy estimator (see Table 7, HTsallis_HRenyi_initialization.m, HTsallis_HRenyi_estimation.m). Making use of this approach, for example, the Rényi entropy estimators of Table 2 can be instantly applied for Tsallis entropy estimation.

5. Let $\mathbf{y}_G \in \mathbb{R}^d$ be a normal random variable with the same mean and covariance as $\mathbf{y}$:

$$\mathbf{y}_G \sim f_G = N(\mathbb{E}(\mathbf{y}), cov(\mathbf{y})). \tag{71}$$

The Shannon entropy of a normal random variable can be explicitly computed

$$H(\mathbf{y}_G) = \frac{1}{2}\log\left[(2\pi e)^d \det(cov(\mathbf{y}))\right], \tag{72}$$

moreover, $H(\mathbf{y})$ equals to $H(\mathbf{y}_G)$ minus the Kullback-Leibler divergence [see Eq. (36)] of $\mathbf{y} \sim f$ and $f_G$ [113]:

$$H(\mathbf{y}) = H(\mathbf{y}_G) - D(f, f_G). \tag{73}$$

The associated meta entropy estimator is called 'Shannon_DKL_N' (see Table 7, HShannon_DKL_N_initialization.m, HShannon_DKL_N_estimation.m).

6. If $\mathbf{y} \in [0,1]^d$ ($\sim f$), then the entropy of $\mathbf{y}$ equals to minus the Kullback-Leibler divergence [see Eq. (36)] of $f$ and $f_U$, the uniform distribution on $[0,1]^d$:

$$H(\mathbf{y}) = -D(f, f_U). \tag{74}$$

If $\mathbf{y} \in [\mathbf{a}, \mathbf{b}] = \times_{i=1}^d [a_i, b_i] \subseteq \mathbb{R}^d$ ($\sim f$), then let $\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{d} \sim f'$ be its linearly transformed version to $[0,1]^d$, where $\mathbf{A} = diag\left(\frac{1}{b_i - a_i}\right) \in \mathbb{R}^{d\times d}$, $\mathbf{d} = \left[\frac{a_i}{a_i - b_i}\right] \in \mathbb{R}^d$. Applying the previous result and the entropy transformation rule under linear mappings [14], one obtains that

$$H(\mathbf{y}) = -D(f', f_U) + \log\left[\prod_{i=1}^d (b_i - a_i)\right]. \tag{75}$$

This meta entropy estimation technique is called 'Shannon_DKL_U' in ITE (see Table 7, HShannon_DKL_U_initialization.m, HShannon_DKL_U_estimation.m).

The meta entropy estimator methods in ITE are summarized in Table 7. The calling syntax of the estimators is identical to Example 10, one only has to change the name 'ensemble' to the cost_name of the target estimators, see the last column of the table.

### 3.2.2 Mutual Information Estimators

In this section we are dealing with meta mutual information estimators:

1. As it has been seen in (1), mutual information can be expressed via entropy terms. The corresponding method is available in the ITE package under the name '`Shannon_HShannon`' (see Table 8, `IShannon_HShannon_initialization.m`, `IShannon_HShannon_estimation.m`). As a prototype example for meta mutual information estimator the implementation is provided below:

   **Example 11 (Mutual information estimator (meta: initialization))**

```
function [co] = IShannon_HShannon_initialization(mult)
co.name = 'Shannon_HShannon';      %name of the estimator: 'Shannon_HShannon'
co.mul = mult;                     %set the importance of multiplicative factors
co.member_name = 'Shannon_kNN_k';  %method used for entropy estimation: 'Shannon_kNN_k'
co.member_co = H_initialization(co.member_name,1);%initialize entropy estimation member, mult=1
```

   **Example 12 (Mutual information estimator (meta: estimation))**

```
function [I] = IShannon_HShannon_estimation(Y,ds,co) %samples(Y), component dimensions(ds),
                                                     %initialized estimator (co)
num_of_comps = length(ds);                    %number of components, M
cum_ds = cumsum([1;ds(1:end-1)]);             %starting indices of the components
I = -H_estimation(Y,co.member_co);            %minus the joint entropy, H([y^1;...;y^M]) using the
                                              %initialized H estimator
for k = 1 : num_of_comps                      %add the entropy of the y^m components, H(y^m)
  idx = [cum_ds(k) : cum_ds(k)+ds(k)-1];
  I = I + H_estimation(Y(idx,:),co.member_co);%use the initialized H estimator
end
```

   The usage of the meta mutual information estimators follow the syntax of base mutual information estimators (see Example 4):

   **Example 13 (Mutual information estimator (meta: usage))**

```
>ds = [1;2]; Y=rand(sum(ds),5000);            %generate the data of interest
                                              %(ds(m)=dim(y^m), T=5000)
>mult = 1;                                    %multiplicative constant is important
>co = IShannon_HShannon_initialization(mult); %initialize the mutual information ('I') estimator
                                              %('Shannon_HShannon')
>I = IShannon_HShannon_estimation(Y,ds,co);   %perform mutual information estimation
```

2. The mutual information of complex random variables ($\mathbf{y} \in \mathbb{C}^{d_m}$) can be defined via the Hilbert transformation [Eq. (68)]:

$$I_{\mathbb{C}}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = I\left(\varphi_v\left(\mathbf{y}^1\right),\ldots,\varphi_v\left(\mathbf{y}^M\right)\right). \tag{76}$$

   The relation is realized in ITE by the '`complex`' meta estimator (see Table 8, `Icomplex_initialization.m`, `Icomplex_estimation.m`).

3. The Shannon-, $L_2$-, Tsallis- and Rényi mutual information can be expressed in terms of the corresponding divergence of the joint ($f$) and the product of marginals ($\prod_{m=1}^M f_m$)[12]:

$$I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D\left(f, \prod_{m=1}^M f_m\right), \qquad I_L\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{L}}\left(f, \prod_{m=1}^M f_m\right), \tag{77}$$

$$I_{\mathrm{T},\alpha}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{T},\alpha}\left(f, \prod_{m=1}^M f_m\right), \qquad I_{\mathrm{R},\alpha}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{R},\alpha}\left(f, \prod_{m=1}^M f_m\right). \tag{78}$$

[12] For the definitions of $f$ and $f_m$s, see Eq. (7). The divergence definitions can be found in Eqs. (36), (37), (38) and (40).

Shannon mutual information is a special case of Rényi's and Tsallis' in limit sense:

$$I_{\mathrm{R},\alpha} \xrightarrow{\alpha\to 1} I, \qquad I_{\mathrm{T},\alpha} \xrightarrow{\alpha\to 1} I. \tag{79}$$

The associated Rényi-, $L_2$- and Tsallis meta mutual information estimators are available in ITE using the names 'Renyi_DRenyi', 'L2_DL2' and 'Tsallis_DTsallis' (see Table 8, `IRenyi_DRenyi_initialization.m`, `IRenyi_DRenyi_estimation.m`, `IL2_DL2_initialization.m`, `IL2_DL2_estimation.m`, `ITsallis_DTsallis_initialization.m`, `ITsallis_DTsallis_estimation.m`).

4. [59] has recently defined a novel, robust, copula-based mutual information measure of the random variable $y^m \in \mathbb{R}$ $(m = 1, \ldots, M)$ as the MMD divergence [Eq. (42)] of the joint copula and the M-dimensional uniform distribution on $[0,1]^M$:

$$I_{\mathrm{c}}\left(y^1, \ldots, y^M\right) = D_{\mathrm{MMD}}(P_{\mathbf{Z}}, P_{\mathbf{U}}), \tag{80}$$

where $\mathbf{Z} = \left[F_1\left(y^1\right); \ldots; F_M\left(y^M\right)\right] \in \mathbb{R}^M$ is the joint copula, $F_m$ is the cumulative density function of $y^m$ and $P$ denotes the distribution. The associated meta estimator has the name 'MMD_DMMD' (see Table 8, `IMMD_DMMD_initialization.m`, `IMMD_DMMD_estimation.m`) in ITE.

5. An alternative form of distance covariance [Eq. (31)] in terms of pairwise distances is

$$I_{\mathrm{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{1'},\mathbf{y}^{2'}}\left[\left\|\mathbf{y}^1 - \mathbf{y}^{1'}\right\|_2 \left\|\mathbf{y}^2 - \mathbf{y}^{2'}\right\|_2\right] + \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^{1'}}\left[\left\|\mathbf{y}^1 - \mathbf{y}^{1'}\right\|_2\right]\mathbb{E}_{\mathbf{y}^2,\mathbf{y}^{2'}}\left[\left\|\mathbf{y}^2 - \mathbf{y}^{2'}\right\|_2\right] \tag{81}$$

$$- 2\mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\left[\mathbb{E}_{\mathbf{y}^{1'}}\left\|\mathbf{y}^1 - \mathbf{y}^{1'}\right\|_2 \mathbb{E}_{\mathbf{y}^{2'}}\left\|\mathbf{y}^2 - \mathbf{y}^{2'}\right\|_2\right], \tag{82}$$

where $(\mathbf{y}^1, \mathbf{y}^2)$ and $(\mathbf{y}^{1'}, \mathbf{y}^{2'})$ are i.i.d. variables. The distance covariance can also be extended to semimetric spaces $[(\mathcal{Y}_1, \rho_1), (\mathcal{Y}_2, \rho_2)]$ of negative type [48, 77] (see Def. 2, Section D):

$$I_{\mathrm{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{1'},\mathbf{y}^{2'}}\left[\rho_1\left(\mathbf{y}^1, \mathbf{y}^{1'}\right)\rho_2\left(\mathbf{y}^2, \mathbf{y}^{2'}\right)\right] + \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^{1'}}\left[\rho_1\left(\mathbf{y}^1, \mathbf{y}^{1'}\right)\right]\mathbb{E}_{\mathbf{y}^2,\mathbf{y}^{2'}}\left[\rho_2\left(\mathbf{y}^2, \mathbf{y}^{2'}\right)\right] \tag{83}$$

$$- 2\mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\left(\mathbb{E}_{\mathbf{y}^{1'}}\left[\rho_1\left(\mathbf{y}^1, \mathbf{y}^{1'}\right)\right]\mathbb{E}_{\mathbf{y}^{2'}}\left[\left(\mathbf{y}^2, \mathbf{y}^{2'}\right)\right]\right). \tag{84}$$

Moreover, it has been proved that the distance covariance can be expressed via HSIC [Eq. (13)]:

$$\left[I_{\mathrm{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right)\right]^2 = 4[D_{\mathrm{MMD}}(f, f_1 f_2)]^2 = 4\left[I_{\mathrm{HSIC}}\left(\mathbf{y}^1, \mathbf{y}^2\right)\right]^2, \tag{85}$$

where the kernel $k$ (used in HSIC) is

$$k((\mathbf{u}_1, \mathbf{v}_1), (\mathbf{u}_2, \mathbf{v}_2)) = k_1(\mathbf{u}_1, \mathbf{u}_2)k_2(\mathbf{v}_1, \mathbf{v}_2) \tag{86}$$

with $k_i$ kernels generating [see Eq. (90)] $\rho_i$-s $(i = 1, 2)$. The meta estimator is called 'dCov_IHSIC' in ITE (see `IdCov_IHSIC_initialization.m`, `IdCov_IHSIC_estimation.m`).

The calling syntax of the meta mutual information are identical (and the same as that of the base estimators, see Section 3.1.2), the possible methods are summarized in Table 8. The techniques are identified by their '`cost_name`', see the last column of the table.

### 3.2.3 Divergence Estimators

In this section we focus on meta divergence estimators (Table 9). Our prototype example is the estimation of the symmetrised Kullback-Leibler divergence, the so-called J-distance:

$$D_{\mathrm{J}}(f_1, f_2) = D(f_1, f_2) + D(f_2, f_1). \tag{87}$$

The definition of meta divergence estimators follows the idea of meta entropy and mutual information estimators (see Example 8, 9, 11 and 12). Initialization and estimation of the meta J-distance estimator can be carried out as follows:

**Example 14 (Divergence estimator (meta: initialization))**

| Estimated quantity | Principle | $d_m$ | $M$ | cost_name |
|---|---|---|---|---|
| complex mutual information ($I_{\mathbb{C}}$) | mutual information of a real random vector variable | $\geq 1$ | $\geq 2$ | 'complex' |
| $L_2$ mutual information ($I_L$) | $L_2$-divergence of the joint and the product of marginals | $\geq 1$ | $\geq 2$ | 'L2_DL2' |
| Rényi mutual information ($I_{\mathrm{R},\alpha}$) | Rényi divergence of the joint and the product of marginals | $\geq 1$ | $\geq 2$ | 'Renyi_DRenyi' |
| copula-based kernel dependency ($I_{\mathrm{c}}$) | MMD div. of the joint copula and the uniform distribution | $= 1$ | $\geq 2$ | 'MMD_DMMD' |
| Rényi mutual information ($I_{\mathrm{R},\alpha}$) | minus the Rényi entropy of the joint copula | $= 1$ | $\geq 2$ | 'Renyi_HRenyi' |
| (Shannon) mutual information ($I$) | entropy sum of the components minus the joint entropy | $\geq 1$ | $\geq 2$ | 'Shannon_HShannon' |
| Tsallis mutual information ($I_{\mathrm{T},\alpha}$) | $L_2$-divergence of the joint and the product of marginals | $\geq 1$ | $\geq 2$ | 'Tsallis_DTsallis' |
| distance covariance ($I_{\mathrm{dCov}}$) | pairwise distances, equivalence to HSIC | $\geq 1$ | $= 2$ | 'dCov_IHSIC' |

Table 8: Mutual information estimators (meta). Third column: dimension constraint ($d_m$; $\mathbf{y}^m \in \mathbb{R}^{d_m}$). Fourth column: constraint for the number of components ($M$; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$).

```
function [co] = DJdistance_initialization(mult)
co.name = 'Jdistance';           %name of the estimator: 'Jdistance'
co.mult = mult;                  %set whether multiplicative constant is important
co.member_name = 'Renyi_kNN_k';  %method used for Kullback-Leibler divergence estimation
co.member_co = D_initialization(co.member_name,mult); %initialize the Kullback-Leibler divergence
                                                      %estimator
```

**Example 15 (Divergence estimator (meta: estimation))**

```
function [D_J] = DJdistance_estimation(X,Y,co)
D_J = D_estimation(X,Y,co.member_co) + D_estimation(Y,X,co.member_co); %definition of J-distance
```

Having defined the J-distance estimator, the calling syntax is completely analogous to base estimators (see Example 5).

**Example 16 (Divergence estimator (meta: usage))**

```
>Y1 = rand(3,1000); Y2 = rand(3,2000); %generate the data of interest (d=3, T_1=1000, T_2=2000)
>mult = 1;                            %multiplicative constant is important
>co = DJdistance_initialization(mult); %initialize the divergence ('D') estimator ('Jdistance')
>D = DJdistance_estimation(Y1,Y2,co);  %perform divergence estimation
```

Further meta divergence estimators of ITE are the following:

1. As is well-known the Kullback-Leibler divergence can be expressed in terms of cross-entropy (see Eq. (65)) and entropy:

$$D(f_1, f_2) = C_{\mathrm{CE}}(f_1, f_2) - H(f_1). \tag{88}$$

   The associated meta divergence estimator is called 'KL_CE_HShannon'.

2. As it has been proved recently [48, 77], the energy distance [Eq. (51)] is closely related to MMD [Eq. (42)]:

$$D_{\mathrm{EnDist}}(f_1, f_2) = 2\left[D_{\mathrm{MMD}}(f_1, f_2)\right]^2, \tag{89}$$

   where the kernel $k$ (used in MMD) generates the semimetric $\rho$ (used in energy distance), i.e.,

$$\rho(\mathbf{u}, \mathbf{v}) = k(\mathbf{u}, \mathbf{u}) + k(\mathbf{v}, \mathbf{v}) - 2k(\mathbf{u}, \mathbf{v}). \tag{90}$$

   The name of the associated meta estimator is 'EnergyDist_DMMD'.

The calling form the meta divergence estimators is uniform, one only has to change in Example 16 the cost_name to the value in the last column of Table 9.

### 3.2.4 Association Estimators

One can define and use meta association estimators completely analogously to meta mutual information estimators (see Section 3.2.2).

| Estimated quantity | Principle | $d$ | cost_name |
|---|---|---|---|
| J-distance ($D_{\mathrm{J}}$) | symmetrised Kullback-Leibler divergence | $d \geq 1$ | 'Jdistance' |
| Kullback-Leibler divergence ($D$) | difference of cross-entropy and entropy | $d \geq 1$ | 'KL_CCE_HShannon' |
| Energy distance ($D_{\mathrm{EnDist}}$) | pairwise distances, equivalence to MMD | $d \geq 1$ | 'EnergyDist_DMMD' |

Table 9: Divergence estimators (meta). Third column: dimension ($d$) constraint.

### 3.2.5 Cross Estimators

One can define and use meta cross estimators completely analogously to meta divergence estimators (see Section 3.2.3).

## 3.3 Uniform Syntax of the Estimators

The modularity of the ITE package in terms of (i) the definition and usage of the base/meta entropy, mutual information, divergence, association and cross estimators, and the possibility to (ii) simple embed novel estimators can be assured by following the templates:

1. Initialization:

**Template 1 (Entropy estimator: initialization)**

```
function [co] = H<cost_name>_initialization(mult)
co.name = <cost_name>;
co.mult = mult;
...
```

**Template 2 (Mutual information estimator: initialization)**

```
function [co] = I<cost_name>_initialization(mult)
co.name = <cost_name>
co.mult = mult;
...
```

**Template 3 (Divergence estimator: initialization)**

```
function [co] = D<cost_name>_initialization(mult)
co.name = <cost_name>
co.mult = mult;
...
```

**Template 4 (Association estimator: initialization)**

```
function [co] = A<cost_name>_initialization(mult)
co.name = <cost_name>
co.mult = mult;
...
```

**Template 5 (Cross estimator: initialization)**

```
function [co] = C<cost_name>_initialization(mult)
co.name = <cost_name>
co.mult = mult;
...
```

2. Estimation:

**Template 6 (Entropy estimator: estimation)**

```
function [H] = H<cost_name>_estimation(Y,co)
...
```

**Template 7 (Mutual information estimator: estimation)**

```
function [I] = I<cost_name>_estimation(Y,ds,co)
...
```

**Template 8 (Divergence estimator: estimation)**

```
function [D] = D<cost_name>_estimation(Y1,Y2,co)
...
```

**Template 9 (Association estimator: estimation)**

```
function [A] = A<cost_name>_estimation(Y,ds,co)
...
```

**Template 10 (Cross estimator: estimation)**

```
function [C] = C<cost_name>_estimation(Y1,Y2,co)
...
```

The unified implementation in the ITE toolbox, makes it possible to use high-level initialization and estimation of the information theoretical quantities. The corresponding functions are

- for initialization: `H_initialization.m`, `I_initialization.m`, `D_initialization.m`, `A_initialization.m`, `C_initialization.m`,
- for estimation: `H_estimation.m`, `I_estimation.m`, `D_estimation.m`, `A_estimation.m`, `C_estimation.m`

following the templates:

```
function [co] = H_initialization(cost_name,mult)
function [co] = I_initialization(cost_name,mult)
function [co] = D_initialization(cost_name,mult)
function [co] = A_initialization(cost_name,mult)
function [co] = C_initialization(cost_name,mult)

function [H] = H_estimation(Y,co)
function [I] = I_estimation(Y,ds,co)
function [D] = D_estimation(Y1,Y2,co)
function [A] = A_estimation(Y,ds,co)
function [C] = C_estimation(Y1,Y2,co)
```

Here, the `cost_name` of the entropy, mutual information, divergence, association and cross estimator can be freely chosen in case of

- entropy: from the last column of Table 2 and Table 7.
- mutual information: from the last column of Table 3 and Table 8.

- divergence: from the last column of Table 4 and Table 9.
- association measures: from the last column of Table 5.
- cross quantities: from the last column of Table 6.

By the ITE construction, following for

- entropy: Template 1 (initialization) and Template 6 (estimation),
- mutual information: Template 2 (initialization) and Template 7 (estimation),
- divergence: Template 3 (initialization) and Template 8 (estimation),
- association measure: Template 4 (initialization) and Template 9 (estimation),
- cross quantity: Template 5 (initialization) and Template 10 (estimation),

user-defined estimators can be immediately used. Let us demonstrate idea of the high-level initialization and estimation with a simple example, Example 2 can equivalently be written as:[13]

**Example 17 (Entropy estimation (high-level, usage))**

```
>Y = rand(5,1000);                          %generate the data of interest (d=5, T=1000)
>cost_name = 'Shannon_kNN_k';               %select the objective (Shannon entropy) and
                                            %its estimation method (k-nearest neighbor)
>mult = 1;                                  %multiplicative constant is important
>co = H_initialization(cost_name,mult); %initialize the entropy estimator
>H = H_estimation(Y,co);                    %perform entropy estimation
```

A more complex example family will be presented in Section 4. There, the basic idea will be the following:

1. Independent subspace analysis and its extensions can be formulated as the optimization of information theoretical quantities. There exist many equivalent formulations (objective functions) in the literature, as well as approximate objectives.
2. Choosing a given objective function, estimators following the template syntaxes (Template 1-8) can be used simply by giving their names (`cost_name`).
3. Moreover, the selected estimator can be immediately used in different optimization algorithms of the objective.

# 4 ITE Application in Independent Process Analysis (IPA)

In this section we present an application of the presented estimators in independent subspace analysis (ISA) and its extensions (IPA, independent process analysis). Application of ITE in IPA serves as an illustrative example, how complex tasks formulated as information theoretical optimization problems can be tackled by the estimators detailed in Section 3.

Section 4.1 formulates the problem domain, the independent process analysis (IPA) problem family. In Section 4.2 the solution methods of IPA are detailed. Section 4.3 is about the Amari-index, which can be used to measure the precision of the IPA estimations. The IPA datasets included in the ITE package are introduced in Section 4.4.

## 4.1 IPA Models

In Section 4.1.1 we focus on the simplest linear model, which allows hidden, independent multidimensional sources (subspaces), the so-called independent subspace analysis (ISA) problem. Section 4.1.2 is about the extensions of ISA.

### 4.1.1 Independent Subspace Analysis (ISA)

The ISA problem is defined in the first paragraph. Then (i) the ISA ambiguities, (ii) equivalent ISA objective functions, and (iii) the ISA separation principle are detailed. Thanks to the ISA separation principle one can define many different equivalent *clustering* based ISA objectives and approximations; this is the topic of the next paragraph. ISA optimization methods are presented in the last paragraph.

[13] One can perform mutual information, divergence, association and cross measure estimations similarly.

**The ISA equations** One may think of independent subspace analysis (ISA)[14] [8, 15] as a cocktail party problem, where (i) more than one group of musicians (sources) are playing at the party, and (ii) we have microphones (sensors), which measure the mixed signals emitted by the sources. The task is to estimate the original sources from the mixed observations only.

Formally, let us assume that we have an observation ($\mathbf{x} \in \mathbb{R}^{D_x}$), which is instantaneous linear mixture ($\mathbf{A}$) of the hidden source ($\mathbf{e}$), that is,

$$\mathbf{x}_t = \mathbf{A}\mathbf{e}_t, \tag{91}$$

where

1. the unknown mixing matrix $\mathbf{A} \in \mathbb{R}^{D_x \times D_e}$ has full column rank,
2. source $\mathbf{e}_t = \left[\mathbf{e}_t^1; \ldots; \mathbf{e}_t^M\right] \in \mathbb{R}^{D_e}$ is a vector concatenated (using Matlab notation ';') of components $\mathbf{e}_t^m \in \mathbb{R}^{d_m}$ ($D_e = \sum_{m=1}^{M} d_m$), subject to the following conditions:
   (a) $\mathbf{e}_t$ is assumed to be i.i.d. (independent identically distributed) in time $t$,
   (b) there is at most one Gaussian variable among $\mathbf{e}^m$s; this assumption will be referred to as the 'non-Gaussian' assumption, and
   (c) $\mathbf{e}^m$s are independent, that is $I\left(\mathbf{e}^1, \ldots, \mathbf{e}^M\right) = 0$.

The goal of the ISA problem is to eliminate the effect of the mixing ($\mathbf{A}$) with a suitable $\mathbf{W} \in \mathbb{R}^{D_e \times D_x}$ *demixing matrix* and estimate the original source components $\mathbf{e}^m$s by using observations $\{\mathbf{x}_t\}_{t=1}^T$ only ($\hat{\mathbf{e}} = \mathbf{W}\mathbf{x}$). If all the $\mathbf{e}^m$ source components are one-dimensional ($d_m = 1, \forall m$), then the independent component analyis (ICA) task [37, 9, 10] is recovered. For $D_x > D_e$ the problem is called *undercomplete*, while the case of $D_x = D_e$ is regarded as *complete*.

**The ISA objective function** One may assume without loss of generality in case of $D_x \geq D_e$ for the full column rank matrix $\mathbf{A}$ that it is invertible – by applying principal component analysis (PCA) [30]. The estimation of the demixing matrix $\mathbf{W} = \mathbf{A}^{-1}$ in ISA is equivalent to the minimization of the mutual information between the estimated components ($\mathbf{y}^m$),

$$J_{\mathrm{I}}(\mathbf{W}) = I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) \rightarrow \min_{\mathbf{W} \in GL(D)}, \tag{92}$$

where $\mathbf{y} = \mathbf{W}\mathbf{x}$, $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$, $\mathbf{y}^m \in \mathbb{R}^{d_m}$, $GL(D)$ denotes the set of $D \times D$ sized invertible matrices, and $D = D_e$. The joint mutual information [Eq. (92)] can also be expressed from only *pair-wise* mutual information by recursive methods [14]

$$I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = \sum_{m=1}^{M-1} I\left(\mathbf{y}^m, \left[\mathbf{y}^{m+1}, ..., \mathbf{y}^M\right]\right). \tag{93}$$

Thus, an equivalent information theoretical ISA objective to (92) is

$$J_{\mathrm{Irecursive}}(\mathbf{W}) = \sum_{m=1}^{M-1} I\left(\mathbf{y}^m, \left[\mathbf{y}^{m+1}, ..., \mathbf{y}^M\right]\right) \rightarrow \min_{\mathbf{W} \in GL(D)}. \tag{94}$$

However, since in ISA, it can be assumed without any loss of generality—applying zero mean normalization and PCA—that

- $\mathbf{x}$ and $\mathbf{e}$ are *white*, i.e., their expectation value is zero, and their covariance matrix is the identity matrix ($\mathbf{I}$),
- mixing matrix $\mathbf{A}$ is orthogonal ($\mathbf{A} \in \mathcal{O}^D$), that is $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, and
- the task is complete ($D = D_x = D_e$),

[14] ISA is also called multidimensional ICA, independent feature subspace analysis, subspace ICA, or group ICA in the literature. We will use the ISA abbreviation.

one can restrict the optimization in (92) and (94) to the orthogonal group ($\mathbf{W} \in \mathcal{O}^D$). Under the whiteness assumption, well-known identities of mutual information and entropy expressions [14] show that the ISA problem is equivalent to

$$J_{\text{sumH}}(\mathbf{W}) = \sum_{m=1}^{M} H\left(\mathbf{y}^m\right) \to \min_{\mathbf{W}\in\mathcal{O}^D}, \tag{95}$$

$$J_{H,I}(\mathbf{W}) = \sum_{m=1}^{M}\sum_{i=1}^{d_m} H(y_i^m) - \sum_{m=1}^{M} I\left(y_1^m,\ldots,y_{d_m}^m\right) \to \min_{\mathbf{W}\in\mathcal{O}^D}, \tag{96}$$

$$J_{I,I}(\mathbf{W}) = I\left(y_1^1,\ldots,y_{d_M}^M\right) - \sum_{m=1}^{M} I\left(y_1^m,\ldots,y_{d_m}^m\right) \to \min_{\mathbf{W}\in\mathcal{O}^D}, \tag{97}$$

where $\mathbf{y}^m = \left[y_1^m;\ldots;y_{d_m}^m\right]$.

**The ISA ambiguities** Identification of the ISA model is ambiguous. However, the ambiguities of the model are simple: hidden components can be determined up to permutation of the subspaces and up to invertible linear transformations[15] within the subspaces [105].

**The ISA separation principle** One of the most exciting and fundamental hypotheses of the ICA research is the ISA separation principle dating back to 1998 [8]: the ISA task can be solved by ICA preprocessing and then clustering of the ICA elements into statistically independent groups. While the extent of this conjecture, is still an open issue, it has recently been rigorously proven for some distribution types [93]. This principle

- forms the basis of the state-of-the-art ISA algorithms,
- can be used to design algorithms that scale well and efficiently estimate the dimensions of the hidden sources and
- can be extended to different linear-, controlled-, post nonlinear-, complex valued-, partially observed models, as well as to systems with nonparametric source dynamics.

For a recent review on the topic, see [96]. The addressed extension directions are (i) presented in Section 4.1.2, (ii) are covered by the ITE package. In the ITE package the solution of the ISA problem is based on the ISA separation principle, for a demonstration, see `demo_ISA.m`.

**Equivalent clustering based ISA objectives and approximations** According to the ISA separation principle, the solution of the ISA task, i.e., the *global* optimum of the ISA cost function can be found by permuting/clustering the ICA elements into statistically independent groups. Using the concept of demixing matrices, it is sufficient to explore forms

$$\mathbf{W}_{\text{ISA}} = \mathbf{P}\mathbf{W}_{\text{ICA}}, \tag{98}$$

where (i) $\mathbf{P} \in \mathbb{R}^{D\times D}$ is a permutation matrix ($\mathbf{P} \in \mathcal{P}^D$) to be determined, (ii) $\mathbf{W}_{\text{ICA}}$ and $\mathbf{W}_{\text{ISA}}$ is the ICA and ISA demixing matrix, respectively. Thus, assuming that the ISA separation principle holds, and since permuting does not alter the ICA objective [see, e.g., the first term in (96) and (97)], the ISA problem is equivalent to

$$J_{\text{I}}(\mathbf{P}) = I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}, \tag{99}$$

$$J_{\text{Irecursive}}(\mathbf{P}) = \sum_{m=1}^{M-1} I\left(\mathbf{y}^m,\left[\mathbf{y}^{m+1},...,\mathbf{y}^M\right]\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}, \tag{100}$$

$$J_{\text{sumH}}(\mathbf{P}) = \sum_{m=1}^{M} H\left(\mathbf{y}^m\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}, \tag{101}$$

$$J_{\text{sum-I}}(\mathbf{P}) = -\sum_{m=1}^{M} I\left(y_1^m,...,y_{d_m}^M\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}. \tag{102}$$

[15] The condition of invertible linear transformations simplifies to orthogonal transformations for the 'white' case.

Let us note that if our observations are generated by an ISA model then—unlike in the ICA task when $d_m = 1$ $(\forall m)$—pairwise independence is *not* equivalent to mutual independence [10]. However, minimization of the pairwise dependence of the estimated subspaces

$$J_{\text{Ipairwise}}(\mathbf{P}) = \sum_{m_1 \neq m_2} I\left(\mathbf{y}^{m_1}, \mathbf{y}^{m_2}\right) \rightarrow \min_{\mathbf{P} \in \mathcal{P}^D} \tag{103}$$

is an efficient approximation in many situations. An alternative approximation is to consider only the pairwise dependence of the coordinates belonging to different subspaces:

$$J_{\text{Ipairwise1d}}(\mathbf{P}) = \sum_{m_1, m_2 = 1; m_1 \neq m_2}^{M} \sum_{i_1=1}^{d_{m_1}} \sum_{i_2=1}^{d_{m_2}} I\left(y_{i_1}^{m_1}, y_{i_2}^{m_2}\right) \rightarrow \min_{\mathbf{P} \in \mathcal{P}^D}. \tag{104}$$

**ISA optimization methods** Let us fix an ISA objective $J$ [Eq. (99)-(104)]. Our goal is to solve the ISA task, i.e., by the ISA separation principle to find the permutation ($\mathbf{P}$) of the ICA elements minimizing $J$. Below we list a few possibilities for finding $\mathbf{P}$; the methods are covered by ITE.

**Exhaustive way:** The possible number of all permutations, i.e., the number of $\mathbf{P}$ matrices is $D!$, where '!' denotes the factorial function. Considering that the ISA cost function is invariant to the exchange of elements *within* the subspaces (see, e.g., (102)), the number of relevant permutations decreases to $\frac{D!}{\prod_{m=1}^{M} d_m!}$. This number can still be enormous, and the related computations could be formidable justifying searches for efficient approximations that we detail below.

**Greedy way:** Two estimated ICA components belonging to different subspaces are exchanged, if it decreases the value of the ISA cost $J$, as long as such pairs exist [98].

**'Global' way:** Experiences show that greedy permutation search is often sufficient for the estimation of the ISA subspaces. However, if the greedy approach cannot find the true ISA subspaces, then global permutation search method of higher computational burden may become necessary [92]: the cross-entropy solution suggested for the traveling salesman problem [71] can be adapted to this case.

**Spectral clustering:** Now, let us assume that source dimensions ($d_m$) are not known in advance. The lack of such knowledge causes combinatorial difficulty in such a sense that one should try all possible

$$D = d_1 + \ldots + d_M \quad (d_m > 0, M \leq D) \tag{105}$$

dimension allocations to the subspace ($\mathbf{e}^m$) dimensions, where $D$ is the dimension of the hidden source $\mathbf{e}$. The number of these $f(D)$ possibilities grows quickly with the argument, its asymptotic behaviour is known [26, 107]:

$$f(D) \sim \frac{e^{\pi\sqrt{2D/3}}}{4D\sqrt{3}} \tag{106}$$

as $D \rightarrow \infty$. An efficient method with good scaling properties has been put forth in [61] for searching the permutation group for the ISA separation theorem (see Table 10). This approach builds upon the fact that the mutual information between different ISA subspaces $\mathbf{e}^m$ is zero due to the assumption of independence. The method assumes that coordinates of $\mathbf{e}^m$ that fall into the same subspace can be paired by using the *pairwise dependence of the coordinates*. This approaches can be considered as objective (104), with unknown $d_m$ subspace dimensions. One may carry out the clustering by applying spectral approaches (included in ITE), which are (i) robust and (ii) scale excellently, a single general desktop computer can handle about a million observations (in our case estimated ICA elements) within several minutes [114].

### 4.1.2 Extensions of ISA

Below we list some extensions of the ISA model and the ISA separation principle. These different extensions, however, can be used in combinations, too. In all these models, (i) the dimension of the source components ($d_m$) can be different and (ii) one can apply the Amari-index as the performance measure (Section 4.3). The ITE package directly implements the estimation of the following models[16] (the relations of the different models are summarized in Fig.1):

[16] The ITE package includes demonstrations for all the touched directions. The name of the demo files are specified at the end the problem definitions, see paragraphs 'Separation principle'.

| |
|---|
| Construct an undirected graph with nodes corresponding to ICA coordinates and edge weights (similarities) defined by the *pairwise* statistical dependencies, i.e., the mutual information of the estimated ICA elements: $\mathbf{S} = [\hat{I}(\hat{e}_{\mathrm{ICA},i}, \hat{e}_{\mathrm{ICA},j})]_{i,j=1}^{D}$. Cluster the ICA elements, i.e., the nodes using similarity matrix $\mathbf{S}$. |

Table 10: Well-scaling approximation for the permutation search problem in the ISA separation theorem in case of unknown subspace dimensions [`estimate_clustering_UD1_S.m`].

**Linear systems:**

**AR-IPA:**

**Equations, assumptions:** In the AR-IPA (autoregressive-IPA) task [32] ($d_m = 1$, $\forall m$), [62] ($d_m \geq 1$), the traditional *i.i.d.* assumption for the sources is generalized to AR time series: the hidden sources ($\mathbf{s}^m \in \mathbb{R}^{d_m}$) are not necessarily independent in time, only their driving noises ($\mathbf{e}^m \in \mathbb{R}^{d_m}$) are. The observation ($\mathbf{x} \in \mathbb{R}^D$, $D = \sum_{m=1}^{M} d_m$) is an instantaneous linear mixture ($\mathbf{A}$) of the source $\mathbf{s}$:

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad \mathbf{s}_t = \sum_{i=1}^{L_s} \mathbf{F}_i \mathbf{s}_{t-i} + \mathbf{e}_t, \tag{107}$$

where $L_s$ is the order of the AR process, $\mathbf{s}_t = \left[\mathbf{s}_t^1; \ldots; \mathbf{s}_t^M\right]$ and $\mathbf{e}_t = \left[\mathbf{e}_t^1; \ldots; \mathbf{e}_t^M\right] \in \mathbb{R}^D$ denote the hidden sources and the hidden driving noises, respectively. (107) can be rewritten in the following concise form:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \qquad \mathbf{F}[z]\mathbf{s} = \mathbf{e} \tag{108}$$

using the polynomial of the time-shift operator $\mathbf{F}[z] := \mathbf{I} - \sum_{i=1}^{L_s} \mathbf{F}_i z^i \in \mathbb{R}[z]^{D \times D}$ [44]. We assume that

1. polynomial matrix $\mathbf{F}[z]$ is *stable*, that is $\det(\mathbf{F}[z]) \neq 0$, for all $z \in \mathbb{C}$, $|z| \leq 1$,
2. mixing matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$ is invertible ($\mathbf{A} \in GL(D)$),
3. $\mathbf{e}$ satisfies the ISA assumptions (see Section 4.1.1)

**Goal:** The aim of the AR-IPA task is to estimate hidden sources $\mathbf{s}^m$, dynamics $\mathbf{F}[z]$, driving noises $\mathbf{e}^m$ and mixing matrix $\mathbf{A}$ or its $\mathbf{W}$ inverse given observations $\{\mathbf{x}_t\}_{t=1}^{T}$. For the special case of $L_s = 0$, the ISA task is obtained.

**Separation principle:** The AR-IPA estimation can be carried out by (i) applying AR fit to observation $\mathbf{x}$, (ii) followed by ISA on the estimated innovation of $\mathbf{x}$ [32, 62]. Demo: `demo_AR_IPA.m`.

**MA-IPA:**

**Equations, assumptions:** Here, the assumption on *instantaneous* linear mixture of the ISA model is weakened to convolutions. This problem is called moving average independent process analysis (MA-IPA, also known as blind subspace deconvolution) [93]. We describe this task for the undercomplete case. Assume that the convolutive mixture of hidden sources $\mathbf{e}^m \in \mathbb{R}^{d_m}$ is available for observation ($\mathbf{x} \in \mathbb{R}^{D_x}$)

$$\mathbf{x}_t = \sum_{l=0}^{L_e} \mathbf{H}_l \mathbf{e}_{t-l}, \tag{109}$$

where

1. $D_x > D_e$ (undercomplete, $D_e = \sum_{m=1}^{M} d_m$),
2. the polynomial matrix $\mathbf{H}[z] = \sum_{l=0}^{L_e} \mathbf{H}_l z^l \in \mathbb{R}[z]^{D_x \times D_e}$ has a (polynomial matrix) left inverse[17] and
3. source $\mathbf{e} = [\mathbf{e}^1; \ldots; \mathbf{e}^M] \in \mathbb{R}^{D_e}$ satisfies the conditions of ISA.

**Goal:** The goal of this undercomplete MA-IPA problem (uMA-IPA problem, where 'u' stands for undercomplete) is to estimate the original $\mathbf{e}^m$ sources by using observations $\{\mathbf{x}_t\}_{t=1}^{T}$ only. The case $L_e = 0$ corresponds to the ISA task, and in the blind source deconvolution problem [57] $d_m = 1$ ($\forall m$), and $L_e$ is a non-negative integer.

[17] One can show for $D_x > D_e$ that under mild conditions $\mathbf{H}[z]$ has a left inverse with probability 1 [70]; e.g., when the matrix $[\mathbf{H}_0, \ldots, \mathbf{H}_{L_e}]$ is drawn from a continuous distribution.

**Note:** We note that in the ISA task the full column rank of matrix $\mathbf{H}_0$ was presumed, which is equivalent to the assumption that matrix $\mathbf{H}_0$ has left inverse. This left inverse assumption is extended in the uMA-IPA model for the polynomial matrix $\mathbf{H}[z]$.

**Separation principle:**

- By applying temporal concatenation (TCC) on the observation, one can reduce the uMA-IPA estimation problem to ISA [93]. Demo: `demo_uMA_IPA_TCC.m`.
- However, upon applying the TCC technique, the associated ISA problem can easily become 'high dimensional'. This dimensionality problem can be alleviated by the linear prediction approximation (LPA) approach, i.e., AR fit, followed by ISA on the estimation innovation [94]. Demo: `demo_uMA_IPA_LPA.m`.
- In the complete ($D_x = D_e$) case, the $\mathbf{H}[z]$ polynomial matrix does not have (polynomial matrix) left inverse in general. However, provided that the convolution can be represented by an infinite order autoregressive [AR($\infty$)] process, one [85] can construct an efficient estimation method for the hidden components via an asymptotically consistent LPA procedure augmented with ISA. Such AR($\infty$) representation can be guaranteed by assuming the stability of $\mathbf{H}[z]$ [21]. Demo: `demo_MA_IPA_LPA.m`.

**Post nonlinear models:**

**Equations, assumptions:** In the post nonlinear ISA (PNL-ISA) problem [97] the *linear* mixing assumption of the ISA model is alleviated. Assume that the observations ($\mathbf{x} \in \mathbb{R}^D$) are post nonlinear mixtures ($\mathbf{g}(\mathbf{A}\cdot)$) of multidimensional independent sources ($\mathbf{e} \in \mathbb{R}^D$):

$$\mathbf{x}_t = \mathbf{g}(\mathbf{A}\mathbf{e}_t), \tag{110}$$

where the

- unknown function $\mathbf{g} : \mathbb{R}^D \to \mathbb{R}^D$ is a component-wise transformation, i.e, $\mathbf{g}(\mathbf{v}) = [g_1(v_1); \ldots; g_D(v_D)]$ and $\mathbf{g}$ is invertible, and
- mixing matrix $\mathbf{A} \in \mathbb{R}^{D\times D}$ and hidden source $\mathbf{e}$ satisfy the ISA assumptions.

**Goal:** The PNL-ISA problem is to estimate the hidden source components $\mathbf{e}^m$ knowing only the observations $\{\mathbf{x}_t\}_{t=1}^T$. For $d_m = 1$, we get back the PNL-ICA problem [103] (for a review see [38]), whereas '$\mathbf{g}$=identity' leads to the ISA task.

**Separation principle:** the estimation of the PNL-ISA problem can be carried out on the basis of the mirror structure of the task, applying gaussianization followed by linear ISA [97]. Demo: `demo_PNL_ISA.m`.

**Complex models:**

**Equations, assumptions:** One can define the independence, mutual information and entropy of complex random variables via the Hilbert transformation [Eq. (68), (69), (76)]. Having these definitions at hand, the complex ISA problem can be formulated analogously to the real case, the observations ($\mathbf{x}_t \in \mathbb{C}^D$) are generated as the instantaneous linear mixture ($\mathbf{A}$) of the hidden sources ($\mathbf{e}_t$):

$$\mathbf{x}_t = \mathbf{A}\mathbf{e}_t, \tag{111}$$

where

- the unknown $\mathbf{A} \in \mathbb{C}^{D\times D}$ mixing matrix is invertible ($D = \sum_{m=1}^M d_m$),
- $\mathbf{e}_t$ is assumed to be i.i.d. in time $t$,
- $\mathbf{e}^m \in \mathbb{C}^{d_m}$s are independent, that is $I\left(\varphi_v\left(\mathbf{e}^1\right), \ldots, \varphi_v\left(\mathbf{e}^M\right)\right) = 0$.

**Goal:** The goal is to estimate the hidden source $\mathbf{e}$ and the mixing matrix $\mathbf{A}$ (or its $\mathbf{W} = \mathbf{A}^{-1}$ inverse) using the observation $\{\mathbf{x}_t\}_{t=1}^T$. If all the components are one-dimensional ($d_m = 1$, $\forall m$), one obtains the complex ICA problem.

**Separation principle:**

- Supposing that the $\varphi_v(\mathbf{e}^m) \in \mathbb{R}^{2d_m}$ variables are 'non-Gaussian', and exploiting the operation preserving property of the Hilbert transformation the solution of the complex ISA problem can be reduced to a ISA task over the real domain with observation $\varphi_v(\mathbf{x})$ and $M$ pieces of $2d_m$-dimensional hidden components $\varphi_v(\mathbf{e}^m)$. The consideration can be extended to *linear models* including AR, MA, ARMA (autoregressive moving average), ARIMA (integrated ARMA), ... terms [88]. Demo: `demo_complex_ISA.m`.

- Another possible solution is to apply the ISA separation theorem, which remains valid even for complex variables [93]: the solution can be accomplished by complex ICA and clustering of the complex ICA elements. Demo: `demo_complex_ISA_C.m`.

**Controlled models:**

**Equations, assumptions:** In the *ARX-IPA* (ARX – autoregressive with exogenous input) problem [87] the AR-IPA assumption holds (Eq. (107)), but the time evolution of the hidden source $\mathbf{s}$ can be influenced via *control* variable $\mathbf{u}_t \in \mathbb{R}^{D_u}$ through matrices $\mathbf{B}_j \in \mathbb{R}^{D \times D_u}$:

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t \qquad\qquad \mathbf{s}_t = \sum_{i=1}^{L_s} \mathbf{F}_i \mathbf{s}_{t-i} + \sum_{j=1}^{L_u} \mathbf{B}_j \mathbf{u}_{t+1-j} + \mathbf{e}_t. \tag{112}$$

**Goal:** The goal is to estimate the hidden source $\mathbf{s}$, the driving noise $\mathbf{e}$, the parameters of the dynamics and control matrices ($\{\mathbf{F}_i\}_{i=1}^{L_s}$ and $\{\mathbf{B}_j\}_{j=1}^{L_u}$), as well as the mixing matrix $\mathbf{A}$ or its inverse $\mathbf{W}$ by using observations $\mathbf{x}_t$ and controls $\mathbf{u}_t$. In the special case of $L_u = 0$, the ARX-IPA task reduces to AR-IPA.

**Separation principle:** The solution can be reduced to ARX identification followed by ISA [87]. Demo: `demo_ARX_IPA.m`.

**Partially observed models:**

**Equations, assumptions:** In the *mAR-IPA* (mAR – autoregressive with missing values) problem [86], the AR-IPA assumptions (Eq. (107)) are relaxed by allowing a few coordinates of the mixed AR sources $\mathbf{x}_t \in \mathbb{R}^D$ to be *missing* at certain time instants. Formally, we observe $\mathbf{y}_t \in \mathbb{R}^D$ instead of $\mathbf{x}_t$, where 'mask mappings' $\mathcal{M}_t : \mathbb{R}^D \mapsto \mathbb{R}^D$ represent the coordinates and the time indices of the non-missing observations:

$$\mathbf{y}_t = \mathcal{M}_t(\mathbf{x}_t), \qquad\qquad \mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad\qquad \mathbf{s}_t = \sum_{i=1}^{L_s} \mathbf{F}_i \mathbf{s}_{t-i} + \mathbf{e}_t. \tag{113}$$

**Goal:** Our task is the estimation of the hidden source $\mathbf{s}$, its driving noise $\mathbf{e}$, parameters of the dynamics $\mathbf{F}[z]$, mixing matrix $\mathbf{A}$ (or its inverse $\mathbf{W}$) from observation $\{\mathbf{y}_t\}_{t=1}^T$. The special case of '$\mathcal{M}_t$ = identity' corresponds to the AR-IPA task.

**Separation principle:** One can reduce the solution to mAR identification followed by ISA on the estimated innovation process [86]. Demo: `demo_mAR_IPA.m`.

**Models with nonparametric dynamics:**

**Equations, assumptions:** In the *fAR-IPA* (fAR – functional autoregressive) problem [91], the *parametric* assumption for the dynamics of the hidden sources is circumvented by functional AR sources:

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad\qquad \mathbf{s}_t = \mathbf{f}(\mathbf{s}_{t-1}, \ldots, \mathbf{s}_{t-L_s}) + \mathbf{e}_t. \tag{114}$$

**Goal:** The goal is to estimate the hidden sources $\mathbf{s}^m \in \mathbb{R}^{d_m}$ including their dynamics $\mathbf{f}$ and their driving innovations $\mathbf{e}^m \in \mathbb{R}^{d_m}$ as well as mixing matrix $\mathbf{A}$ (or its inverse $\mathbf{W}$) given observations $\{\mathbf{x}_t\}_{t=1}^T$. If we knew the parametric form of $\mathbf{f}$ and if it were linear, then the problem would be AR-IPA.

**Separation principle:** The problem can be solved by nonparametric regression followed by ISA [91]. Demo: `demo_fAR_IPA.m`.

## 4.2 Estimation via ITE

Having (i) the information theoretical estimators (Section 3), (ii) the ISA/IPA problems and separation principles (Section 4.1) at hand, we now detail the solution methods offered by the ITE package. Due to the *separation principles* of the IPA problem family, the solution methods can be implemented in a completely *modular* way; the estimation techniques can be built up from the solvers of the obtained *subproblems*. From developer point of view, this flexibility makes it possible to easily modify/extend the ITE toolbox. For example, (i) in case of ISA, one can select/replace the ICA method and clustering technique applied independently, (ii) in case of AR-IPA one has freedom in chosing/extending the AR identificator and the ISA solver, etc. This is the underlying idea of the solvers offered by the ITE toolbox.

In Section 4.2.1 the solution techniques for the ISA task are detailed. Extensions of the ISA problem are in the focus of Section 4.2.2.

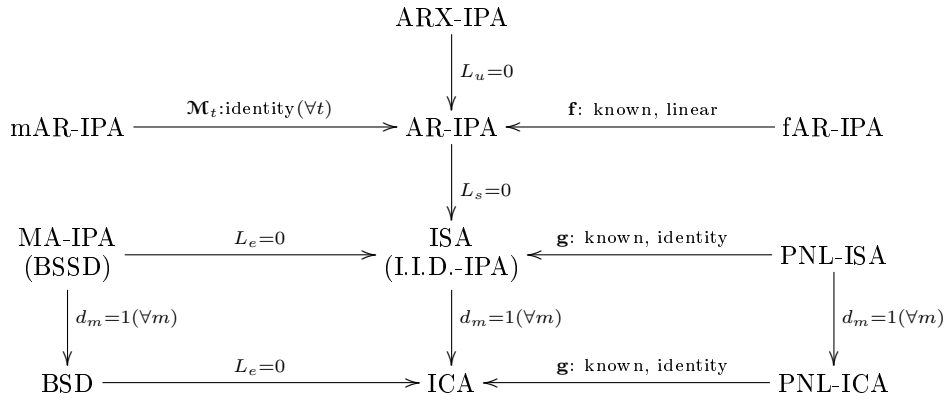



Figure 1: IPA problem family, relations. Arrows point to special cases. For example, 'ISA$\xrightarrow{d_m=1(\forall m)}$ICA' means that ICA is a special case of ISA, when all the source components are one-dimensional.

#### 4.2.1 ISA

As it has been detailed in Section 4.1.1, the ISA problem can be formulated as the optimization of information theoretical objectives (see Eqs. (99), (100), (101), (102), (103), (104)). In the ITE package,

**All the detailed ISA formulations:**

- are available by the appropriate choice of the variable `cost_type` (see Table 11), and
- can be used by *any* entropy/mutual information estimator satisfying the ITE template construction (see Table 2, Table 3, Table 7, Table 8 and Section 3.3).

**The dimension of the subspaces can be given/unknown:** the priori knowledge about the dimension of the subspaces can be conveyed by the variable `unknown_dimensions`. `unknown_dimensions=0` (`=1`) means given $\{d_m\}_{m=1}^M$ subspace dimensions (unknown subspace dimensions, it is sufficient to give $M$, the number of subspaces). In case of

- given subspace dimensions: clustering of the ICA elements can be carried out in ITE by the exhaustive (`opt_type = 'exhaustive'`), greedy (`opt_type = 'greedy'`), or the cross-entropy (`opt_type = 'CE'`) method.
- unknown subspace dimensions: clustering of the ICA elements can be performed by applying spectral clustering. In this case, the clustering is based on the pairwise mutual information of the one-dimensional ICA elements (Table 11) and the objective is (104), i.e., `cost_type = 'Ipairwise1d'`. The ITE package supports 4 different spectral clustering methods/implementations (Table 12):
  - the unnormalized cut method (`opt_type = 'SP1'`), and two normalized cut techniques (`opt_type = 'SP2'` or `opt_type = 'SP3'`) [80, 54, 110] – the implementations are purely Matlab/Octave, and
  - a fast, normalized cut implementation [80, 13] in C++ with compilable mex files (`opt_type = 'NCut'`).

The ISA estimator capable of handling these options is called `estimate_ISA.m`, and is accompanied by the demo file `demo_ISA.m`. Let us take some examples for the parameters to set in `demo_ISA.m`:

**Example 18 (ISA-1)**

- *Goal: the subspace dimensions $\{d_m\}_{m=1}^M$ are known; apply sum of entropy based ISA formulation (Eq. (101)); estimate the entropy via the Rényi entropy using k-nearest neighbors ($S = \{1, \ldots, k\}$); optimize the objective in a greedy way.*
- *Parameters to set:* `unknown_dimensions = 0`; `cost_type = 'sumH'`; `cost_name = 'Renyi_kNN_1tok'`, `opt_type = 'greedy'`.

**Example 19 (ISA-2)**

| Cost function to minimize | Name (`cost_type`) |
|---|---|
| $I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right)$ | `'I'` |
| $\sum_{m=1}^{M} H\left(\mathbf{y}^m\right)$ | `'sumH'` |
| $-\sum_{m=1}^{M} I\left(y_1^m,...,y_{d_m}^M\right)$ | `'sum-I'` |
| $\sum_{m=1}^{M-1} I\left(\mathbf{y}^m,[\mathbf{y}^{m+1},...,\mathbf{y}^M]\right)$ | `'Irecursive'` |
| $\sum_{m1\neq m2} I\left(\mathbf{y}^{m_1},\mathbf{y}^{m_2}\right)$ | `'Ipairwise'` |
| $\sum_{m_1,m_2=1;m_1\neq m_2}^{M}\sum_{i_1=1}^{d_{m_1}}\sum_{i_2=1}^{d_{m_2}} I\left(y_{i_1}^{m_1},y_{i_2}^{m_2}\right)$ | `'Ipairwise1d'` |

Table 11: ISA formulations. $1-4^{th}$ row: equivalent, $5-6^{th}$ row: necessary conditions.

| Optimization technique (`opt_type`) | Principle | Environment |
|---|---|---|
| `'NCut'` | normalized cut | Matlab |
| `'SP1'` | unnormalized cut | Matlab, Octave |
| `'SP2'`, `'SP3'` | 2 normalized cut methods | Matlab, Octave |

Table 12: Spectral clustering optimizers for given number of subspaces ($M$) [`unknown_dimensions=1`]: `clustering_UD1.m`: `estimate_clustering_UD1_S.m`.

- *Goal: the subspace dimensions $\{d_m\}_{m=1}^M$ are known; apply an ISA formulation based on the sum of mutual information within the subspaces (Eq.* (102)*); estimate the mutual information via the KCCA method; optimize the objective in a greedy way.*
- *Parameters to set:* `unknown_dimensions = 0;` `cost_type = 'sum-I';` `cost_name = 'KCCA',` `opt_type = 'greedy'`.

**Example 20 (ISA-3)**

- *Goal: the subspace dimensions are unknown, only $M$, the number of the subspaces is given; the ISA objective is based on the pairwise mutual information of the estimated ICA elements (Eq.* (104)*); estimate the mutual information using the KGV method; optimize the objective via the NCut normalized cut method.*
- *Parameters to set:* `unknown_dimensions = 0;` `cost_type = 'KGV';` `cost_name = 'KGV',` `opt_type = 'NCut'`.

In case of given subspace dimensions, the special structure of the ISA objectives can be taken into account to further increase the efficiency of the **optimization**, i.e., the clustering step. The ITE package realizes this idea:

- In case of (i) one-dimensional mutual information based ISA formulation (Eq. (104)), and (ii) cross-entropy or exhaustive optimization the $\mathbf{S} = [I(\hat{e}_{\mathrm{ICA,i}}, \hat{e}_{\mathrm{ICA,j}})]_{i,j=1}^D$ similarity matrix can be precomputed.
- In case of greedy optimization:
  - upon applying ISA objective (104), the $\mathbf{S} = [I(\hat{e}_{\mathrm{ICA,i}}, \hat{e}_{\mathrm{ICA,j}})]_{i,j=1}^D$ similarity matrix can again be precomputed giving rise to more efficient optimization.
  - ISA formulations (101), (102) are both additive w.r.t. the estimated subspaces. Making use of this special structure of these objective, it is sufficient to recompute the objective only on the touched subspaces while greedily testing a new permutation candidate. Provided that the number of the subspaces ($M$) is high, the decreased computational load of the specialized method is emphasized.
  - objective (103) is pair-additive w.r.t. the subspaces. In this case, it is enough to recompute the objective on the subspaces connected the actual subspace estimates. Again the increased efficiency is striking in case of large number of subspaces.

The general and the recommended (which are chosen by default in the toolbox) ISA optimization methods of ITE are listed Table 13 (greedy), Table 14 (cross-entropy), Table 15 (exhaustive).

**Extending the capabilities of the ITE toolbox**: In case of

| Cost type (`cost_type`) | Recommended/chosen optimizer |
|---|---|
| `'I'`, `'Irecursive'` | `clustering_UD0_greedy_general.m` |
| `'sumH'`, `'sum-I'` | `clustering_UD0_greedy_additive_wrt_subspaces.m` |
| `'Ipairwise'` | `clustering_UD0_greedy_pairadditive_wrt_subspaces.m` |
| `'Ipairwise1d'` | `clustering_UD0_greedy_pairadditive_wrt_coordinates.m` |

Table 13: Recommended/chosen optimizers for given subspace dimensions ($\{d_m\}_{m=1}^M$) [`unknown_dimensions=0`] applying greedy [`opt_type='greedy'`] ISA optimization: `clustering_UD0.m`.

| Cost type (`cost_type`) | Recommended/chosen optimizer |
|---|---|
| `'I'`, `'sumH'`, `'sum-I'`, `'Irecursive'`, `'Ipairwise'` | `clustering_UD0_CE_general.m` |
| `'Ipairwise1d'` | `clustering_UD0_CE_pairadditive_wrt_coordinates.m` |

Table 14: Recommended/chosen optimizers for given subspace dimensions ($\{d_m\}_{m=1}^M$) [`unknown_dimensions=0`] applying cross-entropy [`opt_type='CE'`] ISA optimization: `clustering_UD0.m`.

- known subspaces dimensions ($\{d_m\}_{m=1}^M$): the clustering is carried out in `clustering_UD0.m`. Before clustering, first the importance of the constant multipliers must be set in `set_mult.m`.[18]
  - To add a new ISA formulation (`cost_type`):
    - to be able to carry it out general optimization: it is sufficient to add the new `cost_type` entry to `clustering_UD0.m`, and the computation of the new objective to `cost_general.m`.
    - to be able to perform an existing, specialized (not general) optimization: add the new `cost_type` entry to `clustering_UD0.m`, and the computation of the new objective to the corresponding cost procedure. For example, in case of a new objective being additive w.r.t. subspaces (similarly to (101), (102)) it is sufficient to modify `cost_additive_wrt_subspaces_one_subspace.m` in `cost_additive_wrt_subspaces.m`.
    - to be able to perform a non-existing optimization: add the new `cost_type` entry to `clustering_UD0.m` with the specialized solver.
  - To add a new optimization method (`opt_type`): please follow the 3 examples included in `clustering_UD0.m`.
- unknown subspace dimensions ($M$): `clustering_UD1.m` is responsible for the clustering step. It first computes the $\mathbf{S} = [\hat{I}(\hat{e}_{\mathrm{ICA},i}, \hat{e}_{\mathrm{ICA},j})]_{i,j=1}^D$ similarity matrix, and then performs spectral clustering (see Table 10). To include a new clustering technique, one only has to add it to a new case entry in `estimate_clustering_UD1_S.m`.

#### 4.2.2 Extensions of ISA

Due to the IPA separation principles, the solution of the problem family can be carried out in a *modular* way. The solution of all the presented IPA directions are demonstrated through examples in ITE, the demo files and the actual estimators are listed in Table 16. For the obtained subtasks the ITE package provides many efficient estimators (see Table 17):

**ICA, complex ICA:** The fastICA method [33] and its complex variant [7] is one of the most popular ICA approach, it is available in ITE. See `estimate_ICA.m` and `estimate_complex_ICA.m`.

[18] For example, upon applying objective (101) multiplicative constants are irrelevant (important) in case of equal (different) $d_m$ subspace dimensions.

| Cost type (`cost_type`) | Recommended/chosen optimizer |
|---|---|
| `'I'`, `'sumH'`, `'sum-I'`, `'Irecursive'`, `'Ipairwise'` | `clustering_UD0_exhaustive_general.m` |
| `'Ipairwise1d'` | `clustering_UD0_exhaustive_pairadditive_wrt_coordinates.m` |

Table 15: Recommended/chosen optimizers for given subspace dimensions ($\{d_m\}_{m=1}^M$) [`unknown_dimensions=0`] applying exhaustive [`opt_type='exhaustive'`] ISA optimization: `clustering_UD0.m`.

**AR identification:** Identification of AR processes can be carried in the ITE toolbox in 5 different ways (see `estimate_AR.m`):

- using the online Bayesian technique with normal-inverted Wishart prior [39, 60],
- applying [35]
  - nonlinear least squares estimator based on the subspace representation of the system,
  - exact maximum likelihood optimization using the BFGS (Broyden-Fletcher-Goldfarb-Shannon; or the Newton-Raphson) technique,
  - the combination of the previous two approaches.
- making use of the stepwise least squares technique [53, 75].

**ARX identification:** Identification of ARX processes can be carried out by the D-optimal technique of [60] assuming normal-inverted Wishart prior; see `estimate_ARX_IPA.m`.

**mAR identification:** The

- online Bayesian technique with normal-inverted Wishart prior [39, 60],
- nonlinear least squares [35],
- exact maximum likelihood [35], and
- their combination [35]

are available for the identification of mAR processes; see `estimate_mAR.m`.

**fAR identification:** Identification of fAR processes in ITE can be carried out by the strongly consistent, recursive Nadaraya-Watson estimator [28]; see `estimate_fAR.m`.

**spectral clustering:** The ITE toolbox provides 4 methods to perform spectral clustering (see `estimate_clustering_UD1_S.m`):

- the unnormalized cut method, and two normalized cut techniques [80, 54, 110] – the implemetations are purely Matlab/Octave, and
- a fast, normalized cut implementation [80, 13] in C++ with compilable mex files.

**gaussianization:** Gaussianization of the observations can be carried out by the efficient rank method [116], see `estimate_gaussianization.m`.

**Extending the capabilities of the ITE toolbox**: additional methods for the obtained subtasks can be easily embedded and instantly used in IPA, by simply adding a new 'switch: case' entry to the subtask solvers listed in Table 17.

Beyond the solvers for the IPA subproblems detailed above, the ITE toolbox offers:

- 4 different alternatives for *k-nearest neighbor* estimation (Table 18):
  - exact nearest neighbors: based on fast computation of pairwise distances and C++ partial sort (knn package).
  - exact nearest neighbors: based on fast computation of pairwise distances.
  - exact nearest neighbors: carried out by the `knnsearch` function of the Statistics Toolbox in Matlab.
  - approximate nearest neighbors: implemented by the ANN library.

  The method applied for the estimation can be chosen by setting `co.method` to `'knnFP1'`, `'knnFP2'`, `'knnsearch'`, or `'ANN'`. For examples, please see:
  - `HRenyi_GSF_initialization.m`, `HShannon_kNN_k_initialization.m`, `HRenyi_kNN_1tok_initialization.m`, `HRenyi_kNN_k_initialization.m`, `HRenyi_kNN_S_initialization.m`, `HRenyi_weightedkNN_initialization.m`, `HTsallis_kNN_k_initialization.m`,
  - `DL2_kNN_k_initialization.m`, `DRenyi_kNN_k_initialization.m`, `DTsallis_kNN_k_initialization.m`, `DKL_kNN_kiTi_initialization.m`, `DHellinger_kNN_k_initialization.m`, `DKL_kNN_k_initialization.m`, `DBhattacharyya_kNN_k_initialization.m`,

| IPA model | Reduction | | Demo (Estimator) |
|---|---|---|---|
| | Task1 | Task2 | |
| ISA | ICA | clustering of the ICA elements | demo_ISA.m (estimate_ISA.m) |
| AR-IPA | AR fit | ISA | demo_AR_IPA.m (estimate_AR_IPA.m) |
| ARX-IPA | ARX fit | ISA | demo_ARX_IPA.m (estimate_ARX_IPA.m) |
| mAR-IPA | mAR fit | ISA | demo_mAR_IPA.m (estimate_mAR_IPA.m) |
| complex ISA | Hilbert transformation | real ISA | demo_complex_ISA.m (estimate_complex_ISA.m) |
| complex ISA | complex ICA | clustering of the ICA elements | demo_complex_ISA_C.m (estimate_complex_ISA_C.m) |
| fAR-IPA | nonparametric regression | ISA | demo_fAR_IPA.m (estimate_fAR_IPA.m) |
| (complete) MA-IPA | linear prediction (LPA) | ISA | demo_MA_IPA_LPA.m (estimate_MA_IPA_LPA.m) |
| undercomplete MA-IPA | temporal concatenation (TCC) | ISA | demo_uMA_IPA_TCC.m (estimate_uMA_IPA_TCC.m) |
| undercomplete MA-IPA | linear prediction (LPA) | ISA | demo_uMA_IPA_LPA.m (estimate_uMA_IPA_LPA.m) |
| PNL-ISA | gaussianization | ISA | demo_PNL_ISA.m (estimate_PNL_ISA.m) |

Table 16: IPA separation principles.

| Subtask | Estimator | Method |
|---|---|---|
| ICA | estimate_ICA.m | 'fastICA' |
| complex ICA | estimate_complex_ICA.m | 'fastICA' |
| AR fit (LPA) | estimate_AR.m | 'NIW', 'subspace', 'subspace-LL', 'LL', 'stepwiseLS' |
| ARX fit | estimate_ARX.m | 'NIW' |
| mAR fit | estimate_mAR.m | 'NIW', 'subspace', 'subspace-LL', 'LL' |
| fAR fit | estimate_fAR.m | 'recursiveNW' |
| spectral clustering | estimate_clustering_UD1_S.m | 'NCut', 'SP1', 'SP2', 'SP3' |
| gaussianization | estimate_gaussianization.m | 'rank' |

Table 17: IPA subtasks and estimators.

| co.kNNmethod | Principle | Environment |
|---|---|---|
| 'knnFP1' | exact NNs, fast pairwise distance computation and C++ partial sort | Matlab, Octave |
| 'knnFP2' | exact NNs, fast pairwise distance computation | Matlab, Octave |
| 'knnsearch' | exact NNs, Statistics Toolbox ∈ Matlab | Matlab |
| 'ANN' | approximate NNs, ANN library | Matlab, Octave[a] |

Table 18: k-nearest neighbor (kNN) methods. The main kNN function is `kNN_squared_distances.m`.

[a] See Table 1.

| co.MSTmethod | Method | Environment |
|---|---|---|
| 'MatlabBGL_Prim' | Prim algorithm (MatlabBGL) | Matlab, Octave[a] |
| 'MatlabBGL_Kruskal' | Kruskal algorithm (MatlabBGL) | Matlab, Octave |
| 'pmtk3_Prim' | Prim algorithm (pmtk3) | Matlab, Octave |
| 'pmtk3_Kruskal' | Kruskal algorithm (pmtk3) | Matlab, Octave |

Table 19: Minimum spanning tree (MST) methods. The main MST function is `compute_MST.m`.

[a] See Table 1.

  - `CCE_kNN_k_initialization.m`.

  The central function of kNN computations is `kNN_squared_distances.m`.

- 4 techniques for *minimum spanning tree* computation (Table 19):
  - the two functions of the MatlabBGL library can be envoked by setting `co.STmethod` to `'MatlabBGL_Prim'` or `'MatlabBGL_Kruskal'`.
  - the purely Matlab/Octave implementations based on the pmtk3 toolbox can be called by setting `co.STmethod` to `'pmtk3_Prim'` or `'pmtk3_Kruskal'`.

  For an example, please see `H_Renyi_MST_initialization.m`. The central function for MST computation is `compute_MST.m`.

To **extend** the capabilities of ITE in k-nearest neighbor or minimum spanning tree computation (which is also immediately inherited to entropy, mutual information, divergence, association and cross measure estimation), it sufficient to the add the new method to `kNN_squared_distances.m` or `compute_MST.m`.

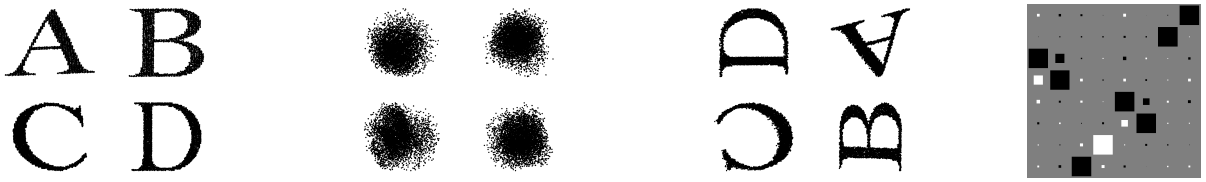
### 4.3 Performance Measure, the Amari-index

Here, we introduce the Amari-index, which can be used to measure the efficiency of the estimators in the ISA problem and its extensions.

Identification of the ISA model is ambiguous. However, the ambiguities of the model are simple: hidden components can be determined up to permutation of the subspaces and up to invertible linear transformations within the subspaces [105]. Thus, in the ideal case, the product of the estimated ISA demixing matrix $\hat{\mathbf{W}}_{\mathrm{ISA}}$ and the ISA mixing matrix $\mathbf{A}$, i.e., matrix

$$\mathbf{G} = \hat{\mathbf{W}}_{\mathrm{ISA}}\mathbf{A} \tag{115}$$

is a block-permutation matrix (also called block-scaling matrix [104]). This property can also be measured for source components with different dimensions by a simple extension [91] of the Amari-index [2], that we present below. Namely, assume that we have a weight matrix $\mathbf{V} \in \mathbb{R}^{M\times M}$ made of positive matrix elements, and a $q \geq 1$ real number. Loosely speaking, we shrink the $d_i \times d_j$ blocks of matrix $\mathbf{G}$ according to the weights of matrix $\mathbf{V}$ and apply the traditional Amari-index for the matrix we obtain. Formally, one can (i) assume without loss of generality that the component dimensions and their estimations are ordered in increasing order ($d_1 \leq \ldots \leq d_M$, $\hat{d}_1 \leq \ldots \leq \hat{d}_M$), (ii) decompose $\mathbf{G}$ into $d_i \times d_j$ blocks



(a) (b) (c) (d)

Figure 2: ISA demonstration (`demo_ISA.m`). (a): hidden components ($\{\mathbf{e}^m\}_{m=1}^M$). (b): observed, mixed signal ($\mathbf{x}$). (c): estimated components ($\{\hat{\mathbf{e}}^m\}_{m=1}^M$). (d): Hinton-diagram: the product of the mixing matrix and the estimated demixing matrix; approximately block-permutation matrix with $2 \times 2$ blocks.

($\mathbf{G} = \left[\mathbf{G}^{ij}\right]_{i,j=1,\ldots,M}$) and define $g^{ij}$ as the $\ell_q$ norm[19] of the elements of the matrix $\mathbf{G}^{ij} \in \mathbb{R}^{d_i \times d_j}$, weighted with $V_{ij}$:

$$g^{ij} = V_{ij} \left( \sum_{k=1}^{d_i} \sum_{l=1}^{d_j} |\left(\mathbf{G}^{ij}\right)_{k,l}|^q \right)^{\frac{1}{q}}. \tag{116}$$

Then the Amari-index with parameters $\mathbf{V}$ can be adapted to the ISA task of possibly different component dimensions as follows

$$r_{\mathbf{V},q}(\mathbf{G}) := \frac{1}{2M(M-1)} \left[ \sum_{i=1}^{M} \left( \frac{\sum_{j=1}^{M} g^{ij}}{\max_j g^{ij}} - 1 \right) + \sum_{j=1}^{M} \left( \frac{\sum_{i=1}^{M} g^{ij}}{\max_i g^{ij}} - 1 \right) \right]. \tag{117}$$

One can see that $0 \leq r_{\mathbf{V},q}(\mathbf{G}) \leq 1$ for any matrix $\mathbf{G}$, and $r_{\mathbf{V},q}(\mathbf{G}) = 0$ if and only if $\mathbf{G}$ is block-permutation matrix with $d_i \times d_j$ sized blocks. $r_{\mathbf{V},q}(\mathbf{G}) = 1$ is in the worst case, i.e, when all the $g^{ij}$ elements are equal. Let us note that this measure (117) is invariant, e.g., for multiplication with a positive constant: $r_{c\mathbf{V}} = r_{\mathbf{V}}$ ($\forall c > 0$). Weight matrix $\mathbf{V}$ can be uniform ($V_{ij} = 1$), or one can use weighing according to the size of the subspaces: $V_{ij} = 1/(d_i d_j)$. The Amari-index [Eq. (117)] is available in the ITE package, see `Amari_index_ISA.m`. The $\mathbf{G}$ global matrix can be visualized by its Hinton-diagram (`hinton_diagram.m`), Fig. 2 provides an illustration. This illustration has been obtained by running `demo_ISA.m`.

The Amari-index can also be used to measure the efficiency of the estimators of the IPA problem family detailed in Section 4.1.2. The demo files in the ITE toolbox (see Table 16) contain detailed examples for the usage of the Amari-index in the extensions of ISA.

## 4.4 Dataset-, Model Generators

One can generate observations from the ISA model and its extensions (Section 4.1.2) by the functions listed in Table 20. The sources/driving datasets can be chosen from many different types in ITE (see `sample_subspaces.m`):

**3D-geom:** In the *3D-geom* test [66] $\mathbf{e}^m$s are random variables uniformly distributed on 3-dimensional geometric forms ($d_m = 3$, $M \leq 6$), see Fig. 3(a). The dataset generator is `sample_subspaces_3D_geom.m`.

**Aw, ABC, GreekABC:** In the $A\omega$ database [98] the distribution of the hidden sources $\mathbf{e}^m$ are uniform on 2-dimensional images ($d_m = 2$) of the English ($M_1 = 26$) and Greek alphabet ($M_2 = 24$). The number of components can be $M = M_1 + M_2 = 50$. Special cases of the database are the *ABC* ($M \leq 26$) [65] and the *GreekABC* ($M \leq 24$) [98] subsets. For illustration, see Fig. 3(d). The dataset generators are called `sample_subspaces_Aw.m`, `sample_subspaces_ABC.m` and `sample_subspaces_GreekABC.m`, respectively.

**mosaic:** The *mosaic* test [95] has 2-dimensional source components ($d_m = 2$) generated from mosaic images. Sources $\mathbf{e}^m$ are generated by sampling 2-dimensional coordinates proportional to the corresponding pixel intensities. In other words, 2-dimensional images are considered as density functions. For illustration, see Fig. 3(h). The dataset generator is `sample_subspaces_mosaic.m`.

**IFS:** Here [97], components $\mathbf{s}^m$ are realizations of IFS[20] based 2-dimensional ($d = 2$) self-similar structures. For all $m$ a $(\{\mathbf{h}_k\}_{k=1,\ldots,K}, \mathbf{p} = (p_1, \ldots, p_K), \mathbf{v}_1\}$ triple is chosen, where

[19] Alternative norms could also be used.

[20] IFS stands for iterated function system.

- $\mathbf{h}_k : \mathbb{R}^2 \to \mathbb{R}^2$ are affine transformations: $\mathbf{h}_k(\mathbf{z}) = \mathbf{C}_k\mathbf{z} + \mathbf{d}_k$ ($\mathbf{C}_k \in \mathbb{R}^{2\times 2}, \mathbf{d}_k \in \mathbb{R}^2$),
- $\mathbf{p}$ is a distribution over the indices $\{1, \ldots, K\}$ ($\sum_{k=1}^{K} p_k = 1, p_k \geq 0$), and
- for the initial value we chose $\mathbf{v}_1 := \left(\frac{1}{2}, \frac{1}{2}\right)$.

In the *IFS* dataset, $T$ samples are generated in the following way: (i) $\mathbf{v}_1$ is given ($t = 1$), (ii) an index $k(t) \in \{1, \ldots, K\}$ is drawn according to the distribution $\mathbf{p}$ and (iii) the next sample is generated as $\mathbf{v}_{t+1} := \mathbf{h}_{k(t)}(\mathbf{v}_t)$. The resulting series $\{\mathbf{v}_1, \ldots, \mathbf{v}_T\}$ was taken as a hidden source component $\mathbf{s}^m$ and this way 9 components ($M = 9$, $D = 18$) were constructed (see Fig. 3(c)). The generator of the dataset is `sample_subspaces_IFS.m`.

**ikeda:** In the *ikeda* test [91], the hidden $\mathbf{s}_t^m = [s_{t,1}^m, s_{t,2}^m] \in \mathbb{R}^2$ sources realize the ikeda map

$$s_{t+1,1}^m = 1 + \lambda_m[s_{t,1}^m \cos(w_t^m) - s_{t,2}^m \sin(w_t^m)], \tag{118}$$

$$s_{t+1,2}^m = \lambda_m[s_{t,1}^m \sin(w_t^m) + s_{t,2}^m \cos(w_t^m)], \tag{119}$$

where $\lambda_m$ is a parameter of the dynamical system and

$$w_t^m = 0.4 - \frac{6}{1 + (s_{t,1}^m)^2 + (s_{t,2}^m)^2}. \tag{120}$$

There are 2 components ($M = 2$) with initial points $\mathbf{s}_1^1 = [20; 20]$, $\mathbf{s}_1^2 = [-100; 30]$ and parameters $\lambda_1 = 0.9994$, $\lambda_2 = 0.998$, see Fig. 3(f) for illustration. Observation can be generated from this dataset using `sample_subspaces_ikeda.m`.

**lorenz:** In the *lorenz* dataset [95], the sources ($\mathbf{s}^m$) correspond to 3-dimensional ($d_m = 3$) deterministic chaotic time series, the so-called Lorenz attractor [47] with different initial points $(x_0, y_0, z_0)$ and parameters ($a$, $b$, $c$). The Lorenz attractor is described by the following ordinary differential equations:

$$\dot{x}_t = a(y_t - x_t), \tag{121}$$

$$\dot{y}_t = x_t(b - z_t) - y_t, \tag{122}$$

$$\dot{z}_t = x_t y_t - c z_t. \tag{123}$$

The differential equations are computed by the explicit Runge-Kutta (4,5) method in ITE. The number of components can be $M = 3$. The dataset generator is `sample_subspaces_lorenz.m`. For illustration, see Fig. 3(g).

**all-k-independent:** In the *all-k-independent* database [65, 92], the $d_m$-dimensional hidden components $\mathbf{v} := \mathbf{e}^m$ are created as follows: coordinates $v_i$ ($i = 1, \ldots, k$) are independent uniform random variables on the set {0,. . . ,k-1}, whereas $v_{k+1}$ is set to $mod(v_1 + \ldots + v_k, k)$. In this construction, every $k$-element subset of $\{v_1, \ldots, v_{k+1}\}$ is made of independent variables and $d_m = k + 1$. The database generator is `sample_subspaces_all_k_independent.m`.

**multiD-geom (multiD$_1$-. . .-D$_M$-geom):** In this dataset $\mathbf{e}^m$s are random variables uniformly distributed on $d_m$-dimensional geometric forms. Geometrical forms were chosen as follows: (i) the surface of the unit ball, (ii) the straight lines that connect the opposing corners of the unit cube, (iii) the broken line between $d_m + 1$ points $\mathbf{0} \to \mathbf{e}_1 \to \mathbf{e}_1 + \mathbf{e}_2 \to \ldots \to \mathbf{e}_1 + \ldots + \mathbf{e}_{d_m}$ (where $\mathbf{e}_i$ is the $i$ canonical basis vector in $\mathbb{R}^{d_m}$, i.e., all of its coordinates are zero except the $i^{th}$, which is 1), and (iv) the skeleton of the unit square. Thus, the number of components $M$ can be equal to 4 ($M \leq 4$), and the dimension of the components ($d_m$) can be scaled. In the *multiD-geom* case the dimensions of the subspaces are equal ($d_1 = \ldots = d_M$); in case of the *multiD$_1$-. . .-D$_M$-geom* dataset, the $d_m$ subspace dimensions can be different. For illustration, see Fig. 3(e). The associated dataset generator is called `sample_subspaces_multiD_geom.m`.

**multiD-spherical (multiD$_1$-. . .-D$_M$-spherical):** In this case hidden sources $\mathbf{e}^m$ are spherical random variables [19]. Since spherical variables assume the form $\mathbf{v} = \rho\mathbf{u}$, where $\mathbf{u}$ is uniformly distributed on the $d_m$-dimensional unit sphere, and $\rho$ is a non-negative scalar random variable independent of $\mathbf{u}$, they can be given by means of $\rho$. 3 pieces of stochastic representations $\rho$ were chosen: $\rho$ was uniform on $[0, 1]$, exponential with parameter $\mu = 1$ and lognormal with parameters $\mu = 0, \sigma = 1$. For illustration, see Fig. 3(b). In this case, the number of component can be 3 ($M \leq 3$) The dimension of the source components ($d_m$) is fixed (can be varied) in the *multiD-spherical* (*multiD$_1$-. . .-D$_M$-spherical*) dataset. Observations can be obtained from these datasets by `sample_subspaces_multiD_spherical.m`.

The datasets and their generators are summarized in Table 21 and Table 22. The `plot_subspaces.m` function can be used to plot the databases (samples/estimations).

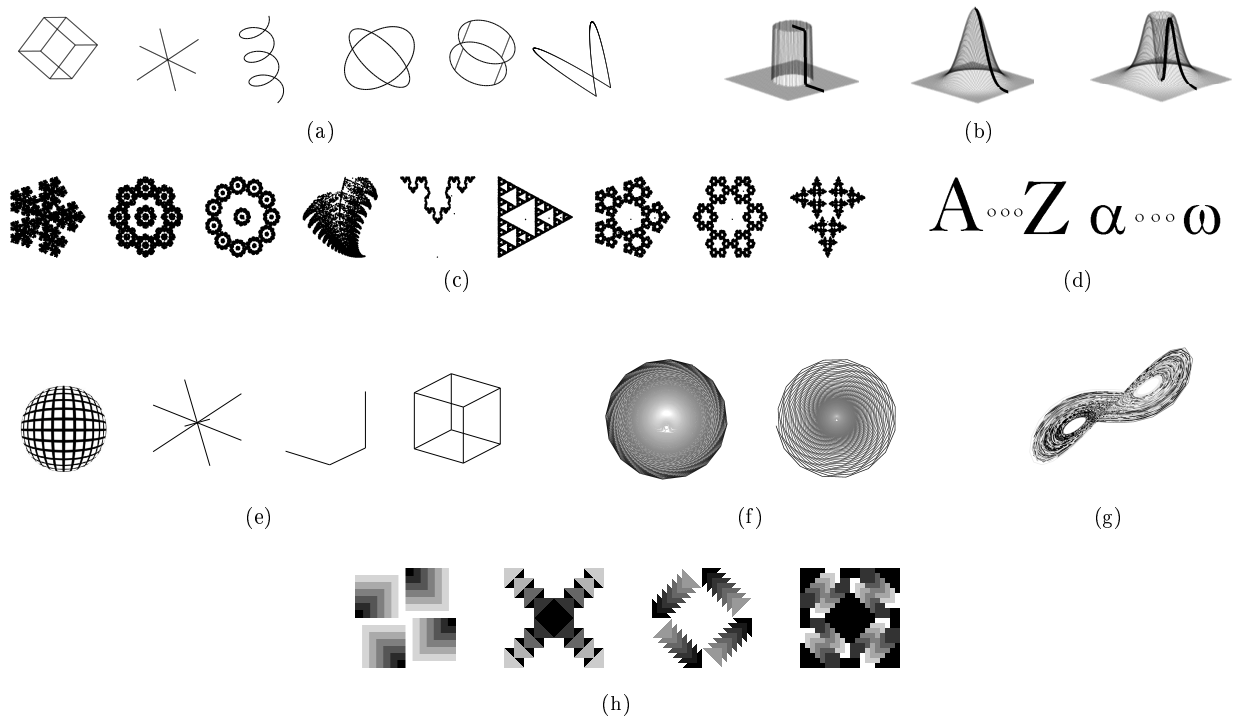



Figure 3: Illustration of the *3D-geom* (a), *multiD-spherical (multi$D_1$-...-$D_M$-spherical)* (b), *IFS* (c), *Aw* (subset on the left: *ABC*, right: *GreekABC*) (d), *multiD-geom (multi$D_1$-...-$D_M$-geom)* (e), *ikeda* (f), *lorenz* (g), and *mosaic* (h) datasets.

| Model | Generator |
|---|---|
| ISA | `generate_ISA.m` |
| complex ISA | `generate_complex_ISA.m` |
| AR-IPA | `generate_AR_IPA.m` |
| ARX-IPA | `generate_ARX_IPA_parameters.m` |
| (u)MA-IPA | `generate_MA_IPA.m` |
| mAR-IPA | `generate_mAR_IPA.m` |
| fAR-IPA | `generate_fAR_IPA.m.m` |

Table 20: IPA model generators. Note: in case of the ARX-IPA model, the observations are generated online in accordance with the online D-optimal ARX identification method.

| Dataset (`data_type`) | Description | Subspace dimensions | # of components | i.i.d. |
|---|---|---|---|---|
| `'3D-geom'` | uniformly distributed (U) on 3D forms | $d_m = 3$ | $M \leq 6$ | Y |
| `'Aw'` | U on English and Greek letters | $d_m = 2$ | $M \leq 50$ | Y |
| `'ABC'` | U on English letters | $d_m = 2$ | $M \leq 26$ | Y |
| `'GreekABC'` | U on Greek letters | $d_m = 2$ | $M \leq 24$ | Y |
| `'mosaic'` | distributed according to mosaic images | $d_m = 2$ | $M \leq 4$ | Y |
| `'IFS'` | self-similar construction | $d_m = 2$ | $M \leq 9$ | N |
| `'ikeda'` | Ikeda map | $d_m = 2$ | $M = 2$ | N |
| `'lorenz'` | Lorenz attractor | $d_m = 3$ | $M \leq 3$ | N |
| `'all-k-independent'` | k-tuples in the subspaces are independent | scalable ($d_m = k + 1$) | $M \geq 1$ | Y |
| `'multid-geom'` | U on $d$-dimensional geometrical forms | scalable ($d = d_m \geq 1$) | $M \leq 4$ | Y |
| `'multid`$_1$`-d`$_2$`-...-d`$_M$`-geom'` | U on $d_m$-dimensional geometrical forms | scalable ($d_m \geq 1$) | $M \leq 4$ | Y |
| `'multid-spherical'` | spherical subspaces | scalable ($d = d_m \geq 1$) | $M \leq 3$ | Y |
| `'multid`$_1$`-d`$_2$`-...-d`$_M$`-spherical'` | spherical subspaces | scalable ($d_m \geq 1$) | $M \leq 3$ | Y |

Table 21: Description of the datasets. Last column: Y – yes, N – no.

| Dataset (`data_type`) | Generator |
|---|---|
| `'3D-geom'` | `sample_subspaces_3D_geom.m` |
| `'Aw'` | `sample_subspaces_Aw.m` |
| `'ABC'` | `sample_subspaces_ABC.m` |
| `'GreekABC'` | `sample_subspaces_GreekABC.m` |
| `'mosaic'` | `sample_subspaces_mosaic.m` |
| `'IFS'` | `sample_subspaces_IFS.m` |
| `'ikeda'` | `sample_subspaces_ikeda.m` |
| `'lorenz'` | `sample_subspaces_lorenz.m` |
| `'all-k-independent'` | `sample_subspaces_all_k_independent.m` |
| `'multid-geom'`, `'multid`$_1$`-d`$_2$`-...-d`$_M$`-geom'` | `sample_subspaces_multiD_geom.m` |
| `'multid-spherical'`, `'multid`$_1$`-d`$_2$`-...-d`$_M$`-spherical'` | `sample_subspaces_multiD_spherical.m` |

Table 22: Generators of the datasets. The high-level sampling function of the datasets is `sample_subspaces.m`.

# 5 Directory Structure of the Package

In this section, we describe the directory structure of the ITE toolbox. Directory

- *code*: code of ITE,
  - *H_I_D_A_C*: entropy, mutual information, divergence, association and cross measure estimators (see Section 3).
    - *base*: contains the base estimators; initialization and estimation functions (see Section 3.1).
    - *meta*: the folder of meta estimators; initialization and estimation functions (see Section 3.2).
    - *utilities*: code shared by *base* and *meta*.
  - *IPA*: application of the information theoretical estimators in ITE (see Section 4):
    - *data_generation*: IPA generators corresponding to different datasets and models.
      - *datasets*: sampling from and plotting of the sources (see Table 21, Table 22, Fig. 3).
      - *models*: IPA model generators, see Table 20.
    - *demos*: IPA demonstrations and estimators, see Table 16 and Table 17.
    - *optimization*: IPA optimization methods (see Table 11, Table 12, Table 13, Table 14, and Table 15).
  - *shared*: code shared by *H_I_D_A_C* and *IPA*.
    - *downloaded*, *embedded*: downloaded and embedded packages (see Section 2).
- *doc*: contains a link to this manual.

# A Citation of the ITE Toolbox

The citing information of the ITE toolbox is provided below in BibTeX format:

```
@ARTICLE{szabo12separation,
  AUTHOR =       {Zolt{\'a}n Szab{\'o} and Barnab{\'a}s P{\'o}czos and Andr{\'a}s L{\H{o}}rincz},
  TITLE =        {Separation Theorem for Independent Subspace Analysis and its Consequences},
  JOURNAL =      {Pattern Recognition},
  YEAR =         {2012},
  volume =       {45},
  issue =        {4},
  pages =        {1782-1791},
}

@ARTICLE{szabo07undercomplete,
  AUTHOR =       {Zolt{\'a}n Szab{\'o} and Barnab{\'a}s P{\'o}czos and Andr{\'a}s L{\H{o}}rincz},
  TITLE =        {Undercomplete Blind Subspace Deconvolution},
  JOURNAL =      {Journal of Machine Learning Research},
  YEAR =         {2007},
  volume =       {8},
  pages =        {1063-1095},
}
```

# B Abbreviations

The abbreviations used in the paper are listed in Table 23.

# C Functions with Octave-Specific Adaptations

Functions with Octave-specific adaptations are summarized in Table 24.

# D Further Definitions

Below, some further definitions are enlisted for the self-containedness of the documentation:

**Definition 1 (measures of concordance [73])** *First, we say that a $C_1$ copula is smaller than the $C_2$ copula ($C_1 \prec C_2$), if*

$$C_1(\mathbf{u}) \leq C_2(\mathbf{u}), \quad \left(\forall \mathbf{u} \in [0,1]^d\right). \tag{124}$$

*This pointwise partial ordering on the set of copulas is called* concordance ordering. *Now, a $\kappa$ numeric measure of association on pairs of random variables ($y^1$, $y^2$ whose joint copula is $C$) is called a* measure of concordance, *if it satisfies the following properties:*

1. *it is defined for every $(y^1, y^2)$ pair of continuous random variables,*
2. *$\kappa\left(y^1, y^2\right) \in [-1, 1]$, $\kappa\left(y^1, y^1\right) = 1$, and $\kappa\left(y^1, -y^1\right) = -1$,*
3. *$\kappa\left(y^1, y^2\right) = \kappa\left(y^2, y^1\right)$,*
4. *if $y^1$ and $y^2$ are independent, then $\kappa\left(y^1, y^2\right) = \kappa(\Pi) = 0$,*
5. *$\kappa\left(-y^1, y^2\right) = \kappa\left(y^1, -y^2\right) = -\kappa\left(y^1, y^2\right)$,*
6. *if $C_1 \prec C_2$, then $\kappa(C_1) \leq \kappa(C_2)$,*[21]

[21] Hence the name concordance ordering.

| Abbreviation | Meaning |
|---|---|
| ANN | approximate nearest neighbor |
| AR | autoregressive |
| ARIMA | integrated ARMA |
| ARMA | autoregressive moving average |
| ARX | AR with exogenous input |
| BFGS | Broyden-Fletcher-Goldfarb-Shannon |
| BSD | blind source deconvolution |
| BSSD | blind subspace deconvolution |
| CDSS | continuously differentiable sample spacing |
| CE | cross-entropy |
| fAR | functional AR |
| GV | generalized variance |
| HS | Hilbert-Schmidt |
| HSIC | Hilbert-Schmidt independence criterion |
| ICA/ISA/IPA | independent component/subspace/process analysis |
| i.i.d. | independent identically distributed |
| IFS | iterated function system |
| IPM | integral probability metrics |
| ITE | information theoretical estimators |
| JFD | joint f-decorrelation |
| KCCA | kernel canonical correlation analysis |
| KDE | kernel density estimation |
| KL | Kullback-Leibler |
| KGV | kernel generalized variance |
| kNN | k-nearest neighbor |
| LPA | linear prediction approximation |
| MA | moving average |
| mAR | AR with missing values |
| MMD | maximum mean discrepancy |
| NIW | normal-inverted Wishart |
| NN | nearest neighbor |
| PCA | principal component analysis |
| PNL | post nonlinear |
| QMI | quadratic mutual information |
| RBF | radial basis function |
| RKHS | reproducing kernel Hilbert space |
| RP | random projection |

Table 23: Abbrevations.

| Function | Role |
|---|---|
| `ITE_install.m` | installation of the ITE package |
| `hinton_diagram.m` | Hinton-diagram |
| `estimate_clustering_UD1_S.m` | spectral clustering |
| `control.m` | D-optimal control |
| `sample_subspaces_lorenz.m` | sampling from the *lorenz* dataset |
| `clinep.m` | the core of the 3D trajectory plot |
| `plot_subspaces_3D_trajectory.m` | 3D trajectory plot |
| `IGV_similarity_matrix.m` | similarity matrix for the GV measure |
| `calculateweight.m` | weight computation in the weighted kNN method |
| `kNN_squared_distances.m` | kNN computation |
| `initialize_Octave_ann_wrapper_if_needed.m` | ann Octave wrapper initialization |
| `IGV_estimation.m` | generalized variance estimation |
| `SpectralClustering.m` | spectral clustering |

Table 24: Functions with Octave-specific adaptations, i.e, the functions calling `working_environment_Matlab.m` (directly).

7. *if* $\left(y^{1,n}, y^{2,n}\right)$ *is a sequence of continuous random variables with copula* $C_n$*, and if* $C_n$ *converges to* $C$ *pointwise, then* $\lim_{n\to\infty} \kappa(C_n) = \kappa(C)$.

**Definition 2 (semimetric space of negative type)** *Let* $\mathcal{Z}$ *be a non-empty set and let* $\rho : \mathcal{Z} \times \mathcal{Z} \to [0, \infty)$ *be a function for which the following properties hold for all* $z, z' \in \mathcal{Z}$*:*

1. $\rho(z, z') = 0$ *if and only if* $z = z'$,
2. $\rho(z, z') = \rho(z', z)$.

*Then* $(Z, \rho)$ *is called a semimetric space.*[22] *A semimetric space is said to be of negative type if*

$$\sum_{i=1}^{T}\sum_{j=1}^{T} a_i a_j \rho(z_i, z_j) \leq 0 \tag{125}$$

*for* $\forall T \geq 2$, $\forall z_1, \ldots, z_T \in \mathcal{Z}$ *and* $\forall a_1, \ldots, a_T \in \mathbb{R}$ *with* $\sum_{i=1}^{T} a_i = 0$.
*Example:*

- *Euclidean spaces are of negative type.*
- *Let* $\mathcal{Z} \subseteq \mathbb{R}^d$ *and* $\rho(z, z') = \|z - z'\|_2^q$. *Then* $(\mathcal{Z}, \rho)$ *is a semimetric space of negative type for* $q \in (0, 2]$.

# E Estimation Formulas – Lookup Table

In this section the underlying entropy (Section E.1), mutual information (Section E.2), divergence (Section E.3), association (Section E.4) and cross (Section E.5) measure computations are summarized *briefly*. This section is considered to be a quick lookup table. For specific details, please see the referred papers (Section 3).

**Notations**: '$*$' denotes transposition. $\mathbf{1}$ ($\mathbf{0}$) stands for the vector whose all elements are equal to 1 (0); $\mathbf{1}_u$, $\mathbf{0}_u$ explicitly indicate the dimension ($u$). The RBF (radial basis function; also called the Gaussian kernel) is defined as

$$k(\mathbf{u}, \mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}}. \tag{126}$$

$tr(\cdot)$ stands for trace. Let $N(\mathbf{m}, \mathbf{\Sigma})$ denote the density function of the normal random variable with mean $\mathbf{m}$ and covariance $\mathbf{\Sigma}$.

$$V_d = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} = \frac{2\pi^{d/2}}{d\Gamma\left(\frac{d}{2}\right)} \tag{127}$$

[22] In contrast to a metric space, the triangle equality is not required.

is the volume of the d-dimensional unit ball. $\psi$ is the digamma function. The scalar product of $\mathbf{A} \in \mathbb{R}^{L_1 \times L_2}$, $\mathbf{B} \in \mathbb{R}^{L_1 \times L_2}$ is $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_i \sum_j A_{ij} B_{ij}$. The Hadamard product of $\mathbf{A} \in \mathbb{R}^{L_1 \times L_2}$, $\mathbf{B} \in \mathbb{R}^{L_1 \times L_2}$ is $(\mathbf{A} \circ \mathbf{B})_{ij} = A_{ij} B_{ij}$. Let $\mathbb{I}$ be the indicator function. Let $y_{(t)}$ denote the order statistics of $\{y_t\}_{t=1}^T$, $(y_t \in \mathbb{R})$, i.e., $y_{(1)} \leq \ldots \leq y_{(T)}$; for $y_{(i)} = y_{(1)}$ $(i < 1)$ and $y_{(i)} = y_{(T)}$ $(i > T)$.

## E.1 Entropy

Notations: Let $\mathbf{Y}_{1:T} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ $(\mathbf{y}_t \in \mathbb{R}^d)$ stand for our samples. Let $\rho_k(t)$ denote the Euclidean distance of the $k^{th}$ nearest neighbor of $\mathbf{y}_t$ in the sample $\mathbf{Y}_{1:T} \backslash \{\mathbf{y}_t\}$. Let $V \subseteq \mathbb{R}^d$ be a finite set, $S, S_1, S_2 \subseteq \{1, \ldots, k\}$ are index sets. $NN_S(V)$ stands for the $S$-nearest neighbor graph on $V$. $NN_S(V_2, V_1)$ denotes the S-nearest (from $V_1$ to $V_2$) neighbor graph. $E$ is the expectation operator.

- `Shannon_kNN_k` [41, 81, 23]:

$$\hat{H}(\mathbf{Y}_{1:T}) = \log(T-1) - \psi(k) + \log(V_d) + \frac{d}{T} \sum_{t=1}^{T} \log\left(\rho_k(t)\right). \tag{128}$$

- `Renyi_kNN_k` [115, 45]:

$$C_{\alpha,k} = \left[\frac{\Gamma(k)}{\Gamma(k+1-\alpha)}\right]^{\frac{1}{1-\alpha}}, \tag{129}$$

$$\hat{I}_\alpha(\mathbf{Y}_{1:T}) = \frac{T-1}{T} V_d^{1-\alpha} C_{\alpha,k}^{1-\alpha} \sum_{t=1}^{T} \frac{[\rho_k(t)]^{d(1-\alpha)}}{(T-1)^\alpha}, \tag{130}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log\left(\hat{I}_\alpha(\mathbf{Y}_{1:T})\right). \tag{131}$$

- `Renyi_kNN_1tok` [66]:

$$S = \{1, \ldots, k\}, \tag{132}$$

$$V = \mathbf{Y}_{1:T}, \tag{133}$$

$$L(V) = \sum_{(\mathbf{u},\mathbf{v}) \in edges(NN_S(V))} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{134}$$

$$c = \lim_{T \to \infty} E_{U_{1:T}, u_t : i.i.d., \sim Uniform([0,1]^d)} \left[\frac{L(U_{1:T})}{T^\alpha}\right], \tag{135}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log\left[\frac{L(V)}{cT^\alpha}\right]. \tag{136}$$

- `Renyi_S` [64]:

$$S \subseteq \{1, \ldots, k\}, k \in S, \tag{137}$$

$$V = \mathbf{Y}_{1:T}, \tag{138}$$

$$L(V) = \sum_{(\mathbf{u},\mathbf{v}) \in edges(NN_S(V))} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{139}$$

$$c = \lim_{T \to \infty} E_{U_{1:T}, u_t : i.i.d., \sim Uniform([0,1]^d)} \left[\frac{L(U_{1:T})}{T^\alpha}\right], \tag{140}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log\left[\frac{L(V)}{cT^\alpha}\right]. \tag{141}$$

- `Renyi_weightedkNN` [83]:

$$k_1 = k_1(T) = \left\lceil 0.1\sqrt{T} \right\rceil, \tag{142}$$

$$k_2 = k_2(T) = \left\lceil 2\sqrt{T} \right\rceil, \tag{143}$$

$$N = \left\lfloor \frac{T}{2} \right\rfloor \tag{144}$$

$$M = T - N, \tag{145}$$

$$V_1 = Y_{1:N}, \tag{146}$$

$$V_2 = Y_{N+1:T}, \tag{147}$$

$$S = \{k_1, \ldots, k_2\}, \tag{148}$$

$$\eta_k = \frac{\beta(k, 1-\alpha)}{\Gamma(1-\alpha)} \frac{1}{N} M^{1-\alpha} V_d^{1-\alpha} \sum_{(\mathbf{u},\mathbf{v}) \in edges(NN_S(V_2, V_1))} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{149}$$

$$\hat{I}_{\alpha,\mathbf{w}} = \sum_{k \in S} w_k \eta_k, \tag{150}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log(\hat{I}_{\alpha,\mathbf{w}}), \tag{151}$$

where the $w_k = w_k(T, d, k_1, k_2)$ weights can be precomputed.

- `Renyi_MST` [115]:

$$V = \mathbf{Y}_{1:T}, \tag{152}$$

$$L(V) = \min_{G \in \text{ spanning trees on } V} \sum_{(\mathbf{u},\mathbf{v}) \in edges(G)} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{153}$$

$$c = \lim_{T \to \infty} E_{U_{1:T}, u_t : i.i.d., \sim Uniform([0,1]^d)} \left[ \frac{L(U_{1:T})}{T^\alpha} \right], \tag{154}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log \left[ \frac{L(V)}{cT^\alpha} \right]. \tag{155}$$

- `Renyi_GSF` [12]:

$$S = \{1, \ldots, k\}, \tag{156}$$

$$V = \mathbf{Y}_{1:T}, \tag{157}$$

$$L(V) = \min_{G \in \text{ spanning forest on } NN_S(V)} \sum_{(\mathbf{u},\mathbf{v}) \in edges(G)} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{158}$$

$$c = \lim_{T \to \infty} E_{U_{1:T}, u_t : i.i.d., \sim Uniform([0,1]^d)} \left[ \frac{L(U_{1:T})}{T^\alpha} \right], \tag{159}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log \left[ \frac{L(V)}{cT^\alpha} \right]. \tag{160}$$

- `Tsallis_kNN_k` [45]:

$$C_{\alpha,k} = \left[ \frac{\Gamma(k)}{\Gamma(k+1-\alpha)} \right]^{\frac{1}{1-\alpha}}, \tag{161}$$

$$\hat{I}_\alpha(\mathbf{Y}_{1:T}) = \frac{T-1}{T} V_d^{1-\alpha} C_{\alpha,k}^{1-\alpha} \sum_{t=1}^{T} \frac{[\rho_k(t)]^{d(1-\alpha)}}{(T-1)^\alpha}, \tag{162}$$

$$\hat{H}_{\mathrm{T},\alpha}(\mathbf{Y}_{1:T}) = \frac{1 - \hat{I}_\alpha(\mathbf{Y}_{1:T})}{\alpha - 1}. \tag{163}$$

- `Shannon_Edgeworth` [31]: Since the Shannon entropy is invariant to additive constants ($H(\mathbf{y}) = H(\mathbf{y}+\mathbf{m})$), one can assume without loss of generality that the expectation of $\mathbf{y}$ is zero. The Edgeworth expansion based estimation is

$$\hat{H}(\mathbf{Y}_{1:T}) = H(\phi_d) - \frac{1}{12}\left[\sum_{i=1}^{d}\left(\kappa^{i,i,i}\right)^2 + 3\sum_{i,j=1;i\neq j}^{d}\left(\kappa^{i,i,j}\right)^2 + \frac{1}{6}\sum_{i,j,k=1;i<j<k}^{d}\left(\kappa^{i,j,k}\right)^2\right], \tag{164}$$

where

$$\mathbf{y}_t = \mathbf{y}_t - \frac{1}{T}\sum_{k=1}^{T}\mathbf{y}_k, (t = 1, \ldots, T) \tag{165}$$

$$\mathbf{\Sigma} = \hat{cov}(\mathbf{Y}_{1:T}) = \frac{1}{T-1}\sum_{t=1}^{T}\mathbf{y}_t(\mathbf{y}_t)^*, \tag{166}$$

$$H(\phi_d) = \frac{1}{2}\log\det(\mathbf{\Sigma}) + \frac{d}{2}\log(2\pi) + \frac{d}{2}, \tag{167}$$

$$\sigma_i = \hat{std}(y^i) = \frac{1}{T-1}\sum_{t=1}^{T}\left(y_t^i\right)^2, \quad (i = 1, \ldots, d) \tag{168}$$

$$\kappa^{ijk} = \hat{E}\left[y^i y^j y^k\right] = \frac{1}{T}\sum_{t=1}^{T} y_t^i y_t^j y_t^k, \quad (i, j, k = 1, \ldots, d) \tag{169}$$

$$\kappa^{i,j,k} = \frac{\kappa^{ijk}}{\sigma_i\sigma_j\sigma_k}. \tag{170}$$

- `Shannon_Voronoi` [50]: Let the Voronoi regions associated to samples $\mathbf{y}_1, \ldots, \mathbf{y}_T$ be denoted by $V_1, \ldots, V_T$ ($V_t \subseteq \mathbb{R}^d$). The estimation is as follows:

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T-K}\sum_{V_i:vol(V_i)\neq\infty}\log\left[T \times vol(V_i)\right], \tag{171}$$

where 'vol' denotes volume, and $K$ is the number of Voronoi regions with finite volume.

- `Shannon_spacing_V` [108]:

$$m = m(T) = \left\lfloor\sqrt{T}\right\rfloor, \tag{172}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\log\left(\frac{T}{2m}\left[y_{(i+m)} - y_{(i-m)}\right]\right). \tag{173}$$

- `Shannon_spacing_Vb` [18]:

$$m = m(T) = \left\lfloor\sqrt{T}\right\rfloor, \tag{174}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T-m}\sum_{t=1}^{T-m}\log\left[\frac{T+1}{m}(y_{(t+m)} - y_{(t)})\right] + \sum_{k=m}^{T}\frac{1}{k} + \log\left(\frac{m}{T+1}\right). \tag{175}$$

- `Shannon_spacing_Vpconst` [55]:

$$m = m(T) = \left\lfloor\sqrt{T}\right\rfloor, \tag{176}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\log\left[\frac{T}{c_t m}(y_{(t+m)} - y_{(t-m)})\right], \tag{177}$$

where

$$c_t = \begin{cases} 1, & 1 \leq t \leq m, \\ 2, & m+1 \leq t \leq T-m, \\ 1 & T-m+1 \leq t \leq T. \end{cases} \tag{178}$$

It can be shown [55] that (173) = (177)+$\frac{2m\log(2)}{T}$.

- `Shannon_spacing_Vplin` [16]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{179}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T} \log\left[\frac{T}{c_t m}(y_{(t+m)} - y_{(t-m)})\right], \tag{180}$$

where

$$c_t = \begin{cases} 1 + \frac{t-1}{m}, & 1 \leq t \leq m, \\ 2, & m+1 \leq t \leq T-m, \\ 1 + \frac{T-t}{m} & T-m+1 \leq t \leq T. \end{cases} \tag{181}$$

- `Shannon_spacing_LL` [11]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{182}$$

$$\bar{y}_{(i)} = \frac{1}{2m+1}\sum_{j=i-m}^{i+m} y_{(j)}, \tag{183}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = -\frac{1}{T}\sum_{t=1}^{T} \log\left[\frac{\sum_{j=i-m}^{i+m}\left(y_{(j)} - \bar{y}_{(i)}\right)(j-i)}{T\sum_{j=i-m}^{i+m}\left(y_{(j)} - \bar{y}_{(i)}\right)^2}\right]. \tag{184}$$

- `Renyi_spacing_V` [111]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{185}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha}\log\left[\frac{1}{T}\sum_{t=1}^{T}\left(\frac{T}{2m}\left[y_{(i+m)} - y_{(i-m)}\right]\right)^{1-\alpha}\right]. \tag{186}$$

- `Renyi_spacing_E` [111]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{187}$$

$$t_1 = \sum_{i=2-m}^{0} \frac{y_{(i+m)} - y_{(i+m-1)}}{2}\left(\sum_{j=1}^{i+m-1}\frac{2}{y_{(j+m)} - y_{(j-m)}}\right)^{\alpha}, \tag{188}$$

$$t_2 = \sum_{i=1}^{T+1-m} \frac{y_{(i)} + y_{(i+m)} - y_{(i-1)} - y_{(i+m-1)}}{2}\left(\sum_{j=i}^{i+m-1}\frac{2}{y_{(j+m)} - y_{(j-m)}}\right)^{\alpha}, \tag{189}$$

$$t_3 = \sum_{i=T+2-m}^{T} \frac{y_{(i)} - y_{(i-1)}}{2}\left(\sum_{j=i}^{T}\frac{2}{y_{(j+m)} - y_{(j-m)}}\right)^{\alpha}, \tag{190}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha}\log\left[\frac{t_1 + t_2 + t_3}{T^{\alpha}}\right]. \tag{191}$$

- `qRenyi_CDSS` [56]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{192}$$

$$\hat{H}_{\mathrm{R},2}(\mathbf{Y}_{1:T}) = -\log \left[ \frac{30}{T(T-m)} \sum_{i=1}^{T-m} \sum_{j=i+1}^{i+m-1} \frac{(y_{(j)} - y_{(i+m)})^2 (y_{(j)} - y_{(i)})^2}{(y_{(i+m)} - y_{(i)})^5} \right]. \tag{193}$$

## E.2 Mutual Information

For an $\mathbf{Y}_{1:T} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ sample set ($\mathbf{y}_t \in \mathbb{R}^d$), let $\hat{F}_m$ denote the empirical estimation of $F_m$, the marginal distribution function of the $m^{th}$ coordinate:

$$\hat{F}_m(y) = \sum_{t=1}^{T} \mathbb{I}_{\{y_t^m \leq y\}}, \tag{194}$$

let the vector of *grades* be defined as

$$\mathbf{U} = \left[ F_1\left(y^1\right); \ldots; F_d\left(y^d\right) \right] \in [0,1]^d, \tag{195}$$

and let its empirical analog, the *ranks* be

$$\hat{U}_{mt} = \hat{F}_m(y_t^m) = \frac{1}{T}(\text{rank of } y_t^m \text{ in } y_1^m, \ldots, y_T^m), \quad (m = 1, \ldots, d). \tag{196}$$

Finally, the empirical copula is defined as

$$\hat{C}_T(\mathbf{u}) := \frac{1}{T} \sum_{t=1}^{T} \prod_{i=1}^{d} \mathbb{I}_{\{\hat{U}_{it} \leq u_i\}}, \quad \left(\mathbf{u} = [u_1; \ldots; u_d] \in [0,1]^d\right), \tag{197}$$

specially

$$\hat{C}_T\left(\frac{i_1}{T}, \ldots, \frac{i_T}{T}\right) = \frac{\#\text{ of } \mathbf{y}\text{-s in the sample with } \mathbf{y} \leq \mathbf{y}_{(i_1,\ldots,i_T)}}{T}, \quad (\forall j, i_j = 1, \ldots, T) \tag{198}$$

where $\mathbf{y}_{(i_1,\ldots,i_T)} = [y_{(i_1)}; \ldots; y_{(i_T)}]$ with $y_{(i_j)}$ order statistics in the $j^{th}$ coordinate.

- `HSIC` [25]:

$$H_{ij} = \delta_{ij} - \frac{1}{T}, \tag{199}$$

$$(\mathbf{K}_m)_{ij} = k_m\left(\mathbf{y}_i^m, \mathbf{y}_j^m\right), \tag{200}$$

$$\hat{I}_{\mathrm{HSIC}}\left(\mathbf{Y}_{1:T}\right) = \frac{1}{T^2} \sum_{u=1}^{M-1} \sum_{v=u+1}^{M} tr(\mathbf{K}_u \mathbf{H} \mathbf{K}_v \mathbf{H}). \tag{201}$$

  Currently, $k_m$-s are RBF-s.

- `KCCA`, `KGV` [4, 93]:

$$\kappa_2 = \frac{\kappa T}{2}, \tag{202}$$

$$\mathbf{K}_m = \left[k_m(\mathbf{y}_i^m, \mathbf{y}_j^m)\right]_{i,j=1,\dots,T}, \tag{203}$$

$$\mathbf{H} = \mathbf{I} - \frac{1}{T}\mathbf{1}\mathbf{1}^*, \tag{204}$$

$$\widetilde{\mathbf{K}}_m = \mathbf{H}\mathbf{K}_m\mathbf{H}, \tag{205}$$

$$\begin{pmatrix} (\widetilde{\mathbf{K}}_1 + \kappa_2\mathbf{I}_T)^2 & \widetilde{\mathbf{K}}_1\widetilde{\mathbf{K}}_2 & \cdots & \widetilde{\mathbf{K}}_1\widetilde{\mathbf{K}}_M \\ \widetilde{\mathbf{K}}_2\widetilde{\mathbf{K}}_1 & (\widetilde{\mathbf{K}}_2 + \kappa_2\mathbf{I}_T)^2 & \cdots & \widetilde{\mathbf{K}}_2\widetilde{\mathbf{K}}_M \\ \vdots & \vdots & & \vdots \\ \widetilde{\mathbf{K}}_M\widetilde{\mathbf{K}}_1 & \widetilde{\mathbf{K}}_M\widetilde{\mathbf{K}}_2 & \cdots & (\widetilde{\mathbf{K}}_M + \kappa_2\mathbf{I}_T)^2 \end{pmatrix} \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_M \end{pmatrix} = \tag{206}$$

$$= \lambda \begin{pmatrix} (\widetilde{\mathbf{K}}_1 + \kappa_2\mathbf{I}_T)^2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (\widetilde{\mathbf{K}}_2 + \kappa_2\mathbf{I})^2 & \cdots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (\widetilde{\mathbf{K}}_M + \kappa_2\mathbf{I})^2 \end{pmatrix} \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_M \end{pmatrix}.$$

Let us write Eq. (206) shortly as $\mathbf{A}\mathbf{c} = \lambda\mathbf{B}\mathbf{c}$. Let the minimal eigenvalue of this generalized eigenvalue problem be $\lambda_{\mathrm{KCCA}}$, and $\lambda_{\mathrm{KGV}} = \frac{\det(\mathbf{A})}{\det(\mathbf{B})}$.

$$\hat{I}_{\mathrm{KCCA}}\left(\mathbf{Y}_{1:T}\right) = -\frac{1}{2}\log(\lambda_{\mathrm{KCCA}}), \tag{207}$$

$$\hat{I}_{\mathrm{KGV}}\left(\mathbf{Y}_{1:T}\right) = -\frac{1}{2}\log(\lambda_{\mathrm{KGV}}). \tag{208}$$

At the moment, $k_m$-s are RBF-s.

- `Hoeffding` [29, 22]: The estimation can be computed as

$$h_2(d) = \left(\frac{2}{(d+1)(d+2)} - \frac{1}{2^d}\frac{d!}{\prod_{i=0}^{d}\left(i+\frac{1}{2}\right)} + \frac{1}{3^d}\right)^{-1}, \tag{209}$$

$$\hat{I}_{\Phi}\left(\mathbf{Y}_{1:T}\right) = \sqrt{h_2(d)\left\{\frac{1}{T^2}\sum_{j=1}^{T}\sum_{k=1}^{T}\prod_{i=1}^{d}\left[1 - \max(\hat{U}_{ij}, \hat{U}_{ik})\right] - \frac{2}{T}\frac{1}{2^d}\sum_{j=1}^{T}\prod_{i=1}^{d}\left(1 - \hat{U}_{ij}^2\right) + \frac{1}{3^d}\right\}}. \tag{210}$$

Under small sample adjustment, one can obtain a similar nice expression:

$$h_2(d,T)^{-1} = \frac{1}{T^2}\sum_{j=1}^{T}\sum_{k=1}^{T}\left[1 - \max\left(\frac{j}{T}, \frac{k}{T}\right)\right]^d - \frac{2}{T}\sum_{j=1}^{T}\left[\frac{T(T-1) - j(j-1)}{2T^2}\right]^d + \frac{1}{3^d}\left[\frac{(T-1)(2T-1)}{2T^2}\right]^d, \tag{211}$$

$$\hat{I}_{\Phi}\left(\mathbf{Y}_{1:T}\right) = \sqrt{h_2(d,T)(t_1 - t_2 + t_3)}, \tag{212}$$

where

$$t_1 = \frac{1}{T^2}\sum_{j=1}^{T}\sum_{k=1}^{T}\prod_{i=1}^{d}\left[1 - \max(\hat{U}_{ij}, \hat{U}_{ik})\right], \quad t_2 = \frac{2}{T}\frac{1}{2^d}\sum_{j=1}^{T}\prod_{i=1}^{d}\left(1 - \hat{U}_{ij}^2 - \frac{1 - \hat{U}_{ij}}{T}\right), \quad t_3 = \frac{1}{3^d}\left[\frac{(T-1)(2T-1)}{2T^2}\right]^d. \tag{213}$$

- `SW1`, `SWinf` [76, 40]:

$$\hat{I}_{\mathrm{SW1}}\left(\mathbf{Y}_{1:T}\right) = \hat{\sigma} = 12\frac{1}{T^2-1}\sum_{i_1=1}^{T}\sum_{i_2=1}^{T}\left|\hat{C}_T\left(\frac{i_1}{T}, \frac{i_2}{T}\right) - \frac{i_1}{T}\frac{i_2}{T}\right|. \tag{214}$$

The $\hat{I}_{\mathrm{SWinf}}$ estimation is performed similarly.

- `QMI_CS_KDE_direct`, `QMI_CS_KDE_iChol`, `QMI_ED_KDE_iChol` [78]:

$$I_{\text{QMI-CS}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \log\left[\frac{L_1 L_2}{(L_3)^2}\right], \tag{215}$$

$$I_{\text{QMI-ED}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = L_1 + L_2 - 2L_3, \tag{216}$$

$$(\mathbf{K}_m)_{ij} = k_m(\mathbf{y}_i^m, \mathbf{y}_j^m), \tag{217}$$

$$\hat{L}_1^{\text{direct}} = \frac{1}{T^2}\left\langle \mathbf{K}_1\mathbf{K}_2\right\rangle, \tag{218}$$

$$\hat{L}_2^{\text{direct}} = \frac{1}{T^4}\left(\mathbf{1}_T^*\mathbf{K}_1\mathbf{1}\right)\left(\mathbf{1}_T^*\mathbf{K}_2\mathbf{1}\right), \tag{219}$$

$$\hat{L}_3^{\text{direct}} = \frac{1}{T^3}\mathbf{1}_T^*\mathbf{K}_1\mathbf{K}_2\mathbf{1}_T, \tag{220}$$

$$\mathbf{K}_m \approx \mathbf{G}_m\mathbf{G}_m^*, \tag{221}$$

$$\hat{L}_1^{\text{iChol}} = \frac{1}{T^2}\mathbf{1}_{d_1}^*\left(\mathbf{G}_1^*\mathbf{G}_2 \circ \mathbf{G}_1^*\mathbf{G}_2\right)\mathbf{1}_{d_2}, \tag{222}$$

$$\hat{L}_2^{\text{iChol}} = \frac{1}{T^4}\left\|\mathbf{1}_T^*\mathbf{G}_1\right\|_2^2\left\|\mathbf{1}_T^*\mathbf{G}_2\right\|_2^2, \tag{223}$$

$$\hat{L}_3^{\text{iChol}} = \frac{1}{T^3}\left(\mathbf{1}_T^*\mathbf{G}_1\right)\left(\mathbf{G}_1^*\mathbf{G}_2\right)\left(\mathbf{G}_2^*\mathbf{1}_T\right), \tag{224}$$

  - `QMI_CS_KDE_direct`:

$$k_m(\mathbf{u},\mathbf{v}) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}} \quad (\forall m), \tag{225}$$

$$\hat{I}_{\text{QMI-CS}}\left(\mathbf{Y}_{1:T}^1,\mathbf{Y}_{1:T}^2\right) = \log\left[\frac{\hat{L}_1^{\text{direct}}\hat{L}_2^{\text{direct}}}{(\hat{L}_3^{\text{direct}})^2}\right]. \tag{226}$$

$$\tag{227}$$

  - `QMI_CS_KDE_iChol`:

$$k_m(\mathbf{u},\mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}} \quad (\forall m), \tag{228}$$

$$\hat{I}_{\text{QMI-CS}}\left(\mathbf{Y}_{1:T}^1,\mathbf{Y}_{1:T}^2\right) = \log\left[\frac{\hat{L}_1^{\text{iChol}}\hat{L}_2^{\text{iChol}}}{(\hat{L}_3^{\text{iChol}})^2}\right]. \tag{229}$$

$$\tag{230}$$

  - `QMI_ED_KDE_iChol`:

$$k_m(\mathbf{u},\mathbf{v}) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^d}e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}} \quad (\forall m), \tag{231}$$

$$\hat{I}_{\text{QMI-ED}}\left(\mathbf{Y}_{1:T}^1,\mathbf{Y}_{1:T}^2\right) = \hat{L}_1^{\text{iChol}} + \hat{L}_2^{\text{iChol}} - 2\hat{L}_3^{\text{iChol}}. \tag{232}$$

`dCov`, `dCor` [102, 99]: The estimation can be carried out by computing only the pairwise distances of the sample points:

$$a_{kl} = \left\|\mathbf{y}_k^1 - \mathbf{y}_l^1\right\|_2^\alpha, \quad \bar{a}_{k\cdot} = \frac{1}{T}\sum_{l=1}^T a_{kl}, \quad \bar{a}_{\cdot l} = \frac{1}{T}\sum_{k=1}^T a_{kl}, \quad \bar{a}_{\cdot\cdot} = \frac{1}{T^2}\sum_{k,l=1}^T a_{kl}, \quad A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}, \tag{233}$$

$$b_{kl} = \left\|\mathbf{y}_k^2 - \mathbf{y}_l^2\right\|_2^\alpha, \quad \bar{b}_{k\cdot} = \frac{1}{T}\sum_{l=1}^T b_{kl}, \quad \bar{b}_{\cdot l} = \frac{1}{T}\sum_{k=1}^T b_{kl}, \quad \bar{b}_{\cdot\cdot} = \frac{1}{T^2}\sum_{k,l=1}^T b_{kl}, \quad B_{kl} = b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}, \tag{234}$$

$$\hat{I}_{\mathrm{dCov}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T}\right)=\frac{1}{T}\sqrt{\sum_{k,l=1}^{T}A_{kl}B_{kl}}=\frac{1}{T}\sqrt{\langle\mathbf{A},\mathbf{B}\rangle}, \tag{235}$$

$$\hat{I}_{\mathrm{dVar}}\left(\mathbf{Y}^1_{1:T}\right)=\frac{1}{T}\sqrt{\sum_{k,l=1}^{T}(A_{kl})^2}=\frac{1}{T}\sqrt{\langle\mathbf{A},\mathbf{A}\rangle}, \tag{236}$$

$$\hat{I}_{\mathrm{dVar}}\left(\mathbf{Y}^2_{1:T}\right)=\frac{1}{T}\sqrt{\sum_{k,l=1}^{T}(B_{kl})^2}=\frac{1}{T}\sqrt{\langle\mathbf{B},\mathbf{B}\rangle}, \tag{237}$$

$$\hat{I}_{\mathrm{dCor}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T}\right)=\begin{cases}\frac{I_{\mathrm{dCov}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T}\right)}{\sqrt{I_{\mathrm{dVar}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^1_{1:T}\right)I_{\mathrm{dVar}}\left(\mathbf{Y}^2_{1:T},\mathbf{Y}^2_{1:T}\right)}}, & \text{if } \hat{I}_{\mathrm{dVar}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^1_{1:T}\right)I_{\mathrm{dVar}}\left(\mathbf{Y}^2_{1:T},\mathbf{Y}^2_{1:T}\right)>0,\\ 0, & \text{otherwise.}\end{cases} \tag{238}$$

### E.3 Divergence

We have $T_1$ and $T_2$ i.i.d. samples from the two distributions ($f_1$, $f_2$) to be compared: $\mathbf{Y}^1_{1:T_1}=\left(\mathbf{y}^1_1,\ldots,\mathbf{y}^1_{T_1}\right)$, $\mathbf{Y}^2_{1:T_2}=\left(\mathbf{y}^2_1,\ldots,\mathbf{y}^2_{T_2}\right)$ ($\mathbf{y}^i_t\in\mathbb{R}^d$). Let $\rho_k(t)$ denote the Euclidean distance of the $k^{th}$ nearest neighbor of $\mathbf{y}^1_t$ in the sample $\mathbf{Y}^1_{1:T_1}\backslash\{\mathbf{y}^1_t\}$, and similarly let $\nu_k(t)$ stand for the Euclidean distance of the $k^{th}$ nearest neighbor of $\mathbf{y}^1_t$ in the sample $\mathbf{Y}^2_{1:T_2}\backslash\{\mathbf{y}^1_t\}$. Let us recall the definitions [Eq. (53), (55)]:

$$D_{\mathrm{temp1}}(\alpha)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^{\alpha}\left[f_2(\mathbf{u})\right]^{1-\alpha}\mathrm{d}\mathbf{u}, \tag{239}$$

$$D_{\mathrm{temp2}}(a,b)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^{a}\left[f_2(\mathbf{u})\right]^{b}f_1(\mathbf{y})\mathrm{d}\mathbf{u}. \tag{240}$$

- `L2_kNN_k` [68, 67, 69]:

$$\hat{D}_{\mathrm{L}}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right)=\sqrt{\frac{1}{T_1V_d}\sum_{t=1}^{T_1}\left[\frac{k-1}{(T_1-1)\rho_k^d(t)}-\frac{2(k-1)}{T_2\nu_k^d(t)}+\frac{(T_1-1)\rho_k^d(t)(k-2)(k-1)}{(T_2)^2\nu_k^{2d}(t)k}\right]}. \tag{241}$$

- `Tsallis_kNN_k` [68, 67]:

$$B_{k,\alpha}=\frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}, \tag{242}$$

$$\hat{D}_{\mathrm{temp1}}(\alpha)=B_{k,\alpha}\frac{(T_1-1)^{1-\alpha}}{(T_2)^{1-\alpha}}\frac{1}{T_1}\sum_{t=1}^{T_1}\left[\frac{\rho_k(t)}{\nu_k(t)}\right]^{d(1-\alpha)}, \tag{243}$$

$$\hat{D}_{\mathrm{T},\alpha}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right)=\frac{1}{\alpha-1}\left[\hat{D}_{\mathrm{temp1}}(\alpha)-1\right]. \tag{244}$$

- `Renyi_kNN_k` [68, 67, 69]:

$$B_{k,\alpha}=\frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}, \tag{245}$$

$$\hat{D}_{\mathrm{temp1}}(\alpha)=B_{k,\alpha}\frac{(T_1-1)^{1-\alpha}}{(T_2)^{1-\alpha}}\frac{1}{T_1}\sum_{t=1}^{T_1}\left[\frac{\rho_k(t)}{\nu_k(t)}\right]^{d(1-\alpha)}, \tag{246}$$

$$\hat{D}_{\mathrm{R},\alpha}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right)=\frac{1}{\alpha-1}\log\left[\hat{D}_{\mathrm{temp1}}(\alpha)\right]. \tag{247}$$

- `MMD_Ustat` [24]:

$$k(\mathbf{u},\mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}}, \tag{248}$$

$$t_1 = \frac{1}{(T_1)^2}\sum_{i,j=1}^{T_1} k\left(\mathbf{y}_i^1,\mathbf{y}_j^1\right), \tag{249}$$

$$t_2 = \frac{1}{(T_2)^2}\sum_{i,j=1}^{T_2} k\left(\mathbf{y}_i^2,\mathbf{y}_j^2\right), \tag{250}$$

$$t_3 = \frac{2}{T_1T_2}\sum_{i=1}^{T_1}\sum_{j=1}^{T_2} k\left(\mathbf{y}_i^1,\mathbf{y}_j^2\right), \tag{251}$$

$$\hat{D}_{\mathrm{MMD}}\left(\mathbf{Y}_{1:T_1}^1,\mathbf{Y}_{1:T_2}^2\right) = \sqrt{t_1+t_2-t_3}. \tag{252}$$

- `MMD_Vstat` [24]:

$$k(\mathbf{u},\mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}}, \tag{253}$$

$$t_1 = \frac{1}{T_1(T_1-1)}\sum_{i=1}^{T_1}\sum_{j=1;j\neq i}^{T_1} k\left(\mathbf{y}_i^1,\mathbf{y}_j^1\right), \tag{254}$$

$$t_2 = \frac{1}{T_2(T_2-1)}\sum_{i=1}^{T_2}\sum_{j=1;j\neq i}^{T_2} k\left(\mathbf{y}_i^2,\mathbf{y}_j^2\right), \tag{255}$$

$$t_3 = \frac{2}{T_1T_2}\sum_{i=1}^{T_1}\sum_{j=1}^{T_2} k\left(\mathbf{y}_i^1,\mathbf{y}_j^2\right), \tag{256}$$

$$\hat{D}_{\mathrm{MMD}}\left(\mathbf{Y}_{1:T_1}^1,\mathbf{Y}_{1:T_2}^2\right) = \sqrt{t_1+t_2-t_3}. \tag{257}$$

- `MMD_online` [24]:

$$T' = \left\lfloor \frac{T_1}{2} \right\rfloor \left( = \left\lfloor \frac{T_2}{2} \right\rfloor \right), \tag{258}$$

$$h((\mathbf{x},\mathbf{y}),(\mathbf{u},\mathbf{v})) = k(\mathbf{x},\mathbf{u}) + k(\mathbf{y},\mathbf{v}) - k(\mathbf{x},\mathbf{v}) - k(\mathbf{y},\mathbf{u}), \tag{259}$$

$$\hat{D}_{\mathrm{MMD}}\left(\mathbf{Y}_{1:T}^1,\mathbf{Y}_{1:T}^2\right) = \frac{1}{T'}\sum_{t=1}^{T'} h\left(\left(\mathbf{y}_{2t-1}^1,\mathbf{y}_{2t-1}^2\right),\left(\mathbf{y}_{2t}^1,\mathbf{y}_{2t}^2\right)\right). \tag{260}$$

  Currently, $k$ is RBF.

- `Hellinger_kNN_k` [63]:

$$B_{k,a,b} = V_d^{-(a+b)}\frac{\Gamma(k)^2}{\Gamma(k-a)\Gamma(k-b)}, \tag{261}$$

$$\hat{D}_{\mathrm{temp2}}(a,b) = (T_1-1)^{-a}(T_2)^{-b}B_{k,a,b}\frac{1}{T_1}\sum_{t=1}^{T_1}[\rho_k(t)]^{-da}[\nu_k(t)]^{-db}, \tag{262}$$

$$\hat{D}_{\mathrm{H}}\left(\mathbf{Y}_{1:T_1}^1,\mathbf{Y}_{1:T_2}^2\right) = \sqrt{1-\hat{D}_{\mathrm{temp2}}\left(-\frac{1}{2},\frac{1}{2}\right)}. \tag{263}$$

- `Bhattacharyya_kNN_k` [63]:

$$B_{k,a,b} = V_d^{-(a+b)} \frac{\Gamma(k)^2}{\Gamma(k-a)\Gamma(k-b)}, \tag{264}$$

$$\hat{D}_{\text{temp2}}(a,b) = (T_1-1)^{-a}(T_2)^{-b} B_{k,a,b} \frac{1}{T_1} \sum_{t=1}^{T_1} [\rho_k(t)]^{-da} [\nu_k(t)]^{-db}, \tag{265}$$

$$\hat{D}_{\text{B}}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = -\log\left[\hat{D}_{\text{temp2}}\left(-\frac{1}{2}, \frac{1}{2}\right)\right]. \tag{266}$$

- `KL_kNN_k` [45, 58, 112]:

$$\hat{D}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = \frac{d}{T_1} \sum_{t=1}^{T_1} \log\left[\frac{\nu_k(t)}{\rho_k(t)}\right] + \log\left(\frac{T_2}{T_1-1}\right). \tag{267}$$

- `KL_kNN_kiTi` [112]:

$$k_1 = k_1(T_1) = \left\lfloor \sqrt{T_1} \right\rfloor, \tag{268}$$

$$k_2 = k_2(T_2) = \left\lfloor \sqrt{T_2} \right\rfloor, \tag{269}$$

$$\hat{D}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = \frac{1}{T_1} \sum_{t=1}^{T_1} \log\left[\frac{k_1}{k_2} \frac{T_2}{T_1-1} \frac{\nu_{k_2}^d(t)}{\rho_{k_1}^d(t)}\right] = \frac{d}{T_1} \sum_{t=1}^{T_1} \log\left[\frac{\nu_{k_2}(t)}{\rho_{k_1}(t)}\right] + \log\left(\frac{k_1}{k_2} \frac{T_2}{T_1-1}\right). \tag{270}$$

- `CS_KDE_iChol`, `ED_KDE_iChol` [78];

$$\mathbf{Z}_{1:2T} = \left[\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right], \tag{271}$$

$$k(\mathbf{u}, \mathbf{v}) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^d} e^{\frac{-\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}}, \tag{272}$$

$$(\mathbf{K})_{ij} = k(\mathbf{z}_i, \mathbf{z}_j), \tag{273}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \in \mathbb{R}^{(2T)\times(2T)}, \tag{274}$$

$$\mathbf{K} \approx \mathbf{G}\mathbf{G}^*, \tag{275}$$

$$D_{\text{CS}}(f_1, f_2) = \log\left[\frac{L_1 L_2}{(L_3)^2}\right], \tag{276}$$

$$D_{\text{ED}}(f_1, f_2) = L_1 + L_2 - 2L_3, \tag{277}$$

$$\mathbf{e}_1 = [\mathbf{1}_T; \mathbf{0}_T], \tag{278}$$

$$\mathbf{e}_2 = [\mathbf{0}_T; \mathbf{1}_T], \tag{279}$$

$$\hat{L}_1 = \frac{1}{T^2}(\mathbf{e}_1^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_1), \tag{280}$$

$$\hat{L}_2 = \frac{1}{T^2}(\mathbf{e}_2^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_2), \tag{281}$$

$$\hat{L}_3 = \frac{1}{T^2}(\mathbf{e}_1^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_2), \tag{282}$$

$$\hat{D}_{\text{CS}}\left(\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right) = \log\left[\frac{\hat{L}_1 \hat{L}_2}{(\hat{L}_3)^2}\right], \tag{283}$$

$$\hat{D}_{\text{ED}}\left(\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right) = \hat{L}_1 + \hat{L}_2 - 2\hat{L}_3. \tag{284}$$

- `EnergyDist`:

$$\hat{D}_{\text{EnDist}}(f_1, f_2) = \frac{2}{T_1 T_2} \sum_{t_1=1}^{T_1} \sum_{t_2=1}^{T_2} \rho\left(\mathbf{y}_{t_1}^1, \mathbf{y}_{t_2}^2\right) - \frac{1}{(T_1)^2} \sum_{t_1=1}^{T_1} \sum_{t_2=1}^{T_1} \rho\left(\mathbf{y}_{t_1}^1, \mathbf{y}_{t_2}^1\right) - \frac{1}{(T_2)^2} \sum_{t_1=1}^{T_2} \sum_{t_2=1}^{T_2} \rho\left(\mathbf{y}_{t_1}^2, \mathbf{y}_{t_2}^2\right). \tag{285}$$

### E.4 Association Measures

We are given $T$ samples from the random variable $\mathbf{y} \in \mathbb{R}^d$ ($\mathbf{Y}_{1:T} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$) and our goal is to estimate the association of its $d_m$-dimensional components ($\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$, $\mathbf{y}^m \in \mathbb{R}^{d_m}$).

- `Spearman1`, `Spearman2`, `Spearman3` [82, 74]: One can arrive at explicit formulas by substituting the empirical copula of $\mathbf{y}$ ($\hat{C}_T$, see Eq. (198)) to the definitions of $\hat{A}_{\rho_i}$-s ($i = 1, 2, 3$; see Eqs. (60), (62), (63)). The resulting nonparametric estimators are

$$\hat{A}_{\rho_1}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_1}(\hat{C}_T) = h_\rho(d)\left[2^d \int_{[0,1]^d} \hat{C}_T(\mathbf{u})\mathrm{d}\mathbf{u} - 1\right] = h_\rho(d)\left[\frac{2^d}{T}\sum_{j=1}^{T}\prod_{i=1}^{d}(1-\hat{U}_{ij}) - 1\right], \tag{286}$$

$$\hat{A}_{\rho_2}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_2}(\hat{C}_T) = h_\rho(d)\left[2^d \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\hat{C}_T(\mathbf{u}) - 1\right] = h_\rho(d)\left[\frac{2^d}{T}\sum_{j=1}^{T}\prod_{i=1}^{d}\hat{U}_{ij} - 1\right], \tag{287}$$

$$\hat{A}_{\rho_3}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_3}(\hat{C}_T) = \frac{\hat{A}_{\rho_1}(\mathbf{Y}_{1:T}) + \hat{A}_{\rho_2}(\mathbf{Y}_{1:T})}{2}, \tag{288}$$

  where $h_\rho(d)$ and $\hat{U}_{ij}$ are defined in Eq. (61) and Eq. (196), respectively.

### E.5 Cross Quantities

We have $T_1$ and $T_2$ i.i.d. samples from the two distributions ($f_1$, $f_2$) to be compared: $\mathbf{Y}^1_{1:T_1} = \left(\mathbf{y}^1_1, \ldots, \mathbf{y}^1_{T_1}\right)$, $\mathbf{Y}^2_{1:T_2} = \left(\mathbf{y}^2_1, \ldots, \mathbf{y}^2_{T_2}\right)$ ($\mathbf{y}^i_t \in \mathbb{R}^d$). Let $\nu_k(t)$ denote the Euclidean distance of the $k^{th}$ nearest neighbor of $\mathbf{y}^1_t$ in the sample $\mathbf{Y}^2_{1:T_2} \backslash \{\mathbf{y}^1_t\}$.

- `CE_kNN_k` [45]:

$$\hat{C}_{\mathrm{CE}}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \log(V_d) + \log(T_2) - \psi(k) + \frac{d}{T_1}\sum_{t=1}^{T_1}\log\left[\nu_k(t)\right]. \tag{289}$$

# References


[1] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66:671–687, 2003.

[2] Shun-ichi Amari, Andrzej Cichocki, and Howard H. Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems (NIPS)*, pages 757–763, 1996.

[3] Rosa I. Arriga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projections. *Machine Learning*, 63:161–182, 2006.

[4] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[5] Michéle. Basseville. Divergence measures for statistical data processing - an annotated bibliography. *Signal Processing*, 2012. To appear. `hal.inria.fr/docs/00/54/23/37/PDF/PI-1961.pdf`.

[6] J. Beirlant, E.J. Dudewicz, L. Győrfi, and E.C. van der Meulen. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6:17–39, 1997.

[7] Ella Bingham and Aapo Hyvärinen. A fast fixed-point algorithm for independent component analysis of complex-valued signals. *International Journal of Neural Systems*, 10(1):1–8, 2000.

[8] Jean-François Cardoso. Multidimensional independent component analysis. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1941–1944, 1998.

[9] Jean-François Cardoso and Antoine Souloumiac. Blind beamforming for non-gaussian signals. *IEE Proceedings F, Radar and Signal Processing*, 140(6):362–370, 1993.

[10] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287–314, 1994.

[11] Juan C. Correa. A new estimator of entropy. *Communications in Statistics - Theory and Methods*, 24:2439–2449, 1995.

[12] Jose A. Costa and Alfred O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transactions on Signal Processing*, 52:2210–2221, 2004.

[13] Timothee Cour, Stella Yu, and Jianbo Shi. Normalized cut segmentation code. Copyright 2004 University of Pennsylvania, Computer and Information Science Department.

[14] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, USA, 1991.

[15] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. Fetal electrocardiogram extraction by source subspace separation. In *IEEE SP/Athos Workshop on Higher-Order Statistics*, pages 134–138, 1995.

[16] Nader Ebrahimi, Kurt Pflughoeft, and Ehsan S. Soofi. Two measures of sample entropy. *Statistics and Probability Letters*, 20:225–234, 1994.

[17] Jan Eriksson. Complex random vectors and ICA models: Identifiability, uniqueness and separability. *IEEE Transactions on Information Theory*, 52(3), 2006.

[18] Bert Van Es. Estimating functionals related to a density by a class of statistics based on spacings. *Scandinavian Journal of Statistics*, 19:61–72, 1992.

[19] Kai-Tai Fang, Samuel Kotz, and Kai Wang Ng. *Symmetric multivariate and related distributions*. Chapman and Hall, 1990.

[20] Peter Frankl and Hiroshi Maehara. The Johnson-Lindenstrauss Lemma and the sphericity of some graphs. *Journal of Combinatorial Theory Series A*, 44(3):355 – 362, 1987.

[21] Wayne A. Fuller. *Introduction to Statistical Time Series*. Wiley-Interscience, 1995.

[22] Sandra Gaißer, Martin Ruppert, and Friedrich Schmid. A multivariate version of Hoeffding's phi-square. *Journal of Multivariate Analysis*, 101:2571–2586, 2010.

[23] M. N. Goria, Nikolai N. Leonenko, V. V. Mergel, and P. L. Novi Inverardi. A new class of random vector entropy estimators and its applications in testing statistical hypotheses. *Journal of Nonparametric Statistics*, 17:277–297, 2005.

[24] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[25] Arthur Gretton, Olivier Bousquet, Alexander Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *International Conference on Algorithmic Learnng Theory (ALT)*, pages 63–78, 2005.

[26] Godfrey H. Hardy and Srinivasa I. Ramanujan. Asymptotic formulae in combinatory analysis. *Proceedings of the London Mathematicl Society*, 17(1):75–115, 1918.

[27] Jan Havrda and František Charvát. Quantification method of classification processes. concept of structural $\alpha$-entropy. *Kybernetika*, 3:30–35, 1967.

[28] Nadine Hilgert and Bruno Portier. Strong uniform consistency and asymptotic normality of a kernel based error density estimator in functional autoregressive models. *Statistical Inference for Stochastic Processes*, 15(2):105–125, 2012.

[29] W. Hoeffding. Massstabinvariante korrelationstheorie. *Schriften des Mathematischen Seminars und des Instituts für Angewandte Mathematik der Universität Berlin*, 5:181–233, 1940.

[30] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[31] Marc Van Hulle. Edgeworth approximation of multivariate differential entropy. *Neural Computation*, 17:1903–1910, 2005.

[32] Aapo Hyvärinen. Independent component analysis for time-dependent stochastic processes. In *International Conference on Artificial Neural Networks (ICANN)*, pages 541–546, 1998.

[33] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[34] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing, 1998*, pages 604–613.

[35] Miguel Jerez, Jose Casals, and Sonia Sotoca. *Signal Extraction for Linear State-Space Models: Including a free MATLAB Toolbox for Time Series Modeling and Decomposition*. LAP LAMBERT Academic Publishing, 2011.

[36] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[37] Christian Jutten and Jeanny Hérault. Blind separation of sources: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[38] Christian Jutten and Juha Karhunen. Advances in blind source separation (BSS) and independent component analysis (ICA) for nonlinear systems. *International Journal of Neural Systems*, 14(5):267–292, 2004.

[39] K. Rao Kadiyala and Sune Karlsson. Numerical methods for estimation and inference in bayesian VAR-models. *Journal of Applied Econometrics*, 12:99–132, 1997.

[40] Sergey Kirshner and Barnabás Póczos. ICA and ISA using Schweizer-Wolff measure of dependence. In *International Conference on Machine Learning (ICML)*, pages 464–471, 2008.

[41] L. F. Kozachenko and Nikolai N. Leonenko. A statistical estimate for the entropy of a random vector. *Problems of Information Transmission*, 23:9–16, 1987.

[42] Solomon Kullback and Richard Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[43] Jan Kybic. High-dimensional mutual information estimation for image registration. In *International Conference on Image Processing (ICIP)*, pages 1779–1782, 2004.

[44] Russell H. Lambert. *Multichannel Blind Deconvolution: FIR matrix algebra and separation of multipath mixtures*. PhD thesis, University of Southern California, 1996.

[45] Nikolai Leonenko, Luc Pronzato, and Vippal Savani. A class of Rényi information estimators for multidimensional densities. *Annals of Statistics*, 36(5):2153–2182, 2008.

[46] Ping Li, Trevor J. Hastie, and Kenneth W. Hastie. Very sparse random projections. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 287–296, 2006.

[47] Edward Norton Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20:130–141, 1963.

[48] Russell Lyons. istance covariance in metric spaces. *Annals of Probability*, 2012. (To appear. `http://php.indiana.edu/~rdlyons/pdf/dcov.pdf`).

[49] Jiří Matoušek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures and Algorithms*, 33(2):142–156, 2008.

[50] Erik Miller. A new class of entropy estimators for multi-dimensional densities. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 297–300, 2003.

[51] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29:429–443, 1997.

[52] Roger B. Nelsen. *An Introduction to Copulas (Springer Series in Statistics)*. Springer, 2006.

[53] Arnold Neumaier and Tapio Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software*, 27(1):27–57, 2001.

[54] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856, 2002.

[55] Hadi Alizadeh Noughabi and Naser Reza Arghami. A new estimator of entropy. *Journal of Iranian Statistical Society*, 9:53–64, 2010.

[56] Umut Ozertem, Ismail Uysal, and Deniz Erdogmus. Continuously differentiable sample-spacing entropy estimation. *IEEE Transactions on Neural Networks*, 19:1978–1984, 2008.

[57] Michael S. Pedersen, Jan Larsen, Ulrik Kjems, and Lucas C. Parra. A survey of convolutive blind source separation methods. In *Springer Handbook of Speech Processing*. Springer, 2007.

[58] Fernando Pérez-Cruz. Estimation of information theoretic measures for continuous random variables. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1257–1264, 2008.

[59] Barnabás Póczos, Zoubin Ghahramani, and Jeff Schneider. Copula-based kernel dependency measures. In *International Conference on Machine Learning (ICML)*, 2012.

[60] Barnabás Póczos and András Lőrincz. Identification of recurrent neural networks by Bayesian interrogation techniques. *Journal of Machine Learning Research*, 10:515–554, 2009.

[61] Barnabás Póczos, Zoltán Szabó, Melinda Kiszlinger, and András Lőrincz. Independent process analysis without a priori dimensional information. In *International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 252–259, 2007.

[62] Barnabás Póczos, Bálint Takács, and András Lőrincz. Independent subspace analysis on innovations. In *European Conference on Machine Learning (ECML)*, pages 698–706, 2005.

[63] Barnabás Póczos, Liang Xiong, Dougal Sutherland, and Jeff Schneider. Support distribution machines. Technical report, Carnegie Mellon University, 2012. `http://arxiv.org/abs/1202.0302`.

[64] Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of Rényi entropy and mutual information based on generalized nearest-neighbor graphs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1849–1857, 2011.

[65] Barnabás Póczos and András Lőrincz. Independent subspace analysis using geodesic spanning trees. In *International Conference on Machine Learning (ICML)*, pages 673–680, 2005.

[66] Barnabás Póczos and András Lőrincz. Independent subspace analysis using k-nearest neighborhood estimates. In *International Conference on Artificial Neural Networks (ICANN)*, pages 163–168, 2005.

[67] Barnabás Póczos and Jeff Schneider. On the estimation of $\alpha$-divergences. In *International conference on Artificial Intelligence and Statistics (AISTATS)*, pages 609–617, 2011.

[68] Barnabás Póczos, Zoltán Szabó, and Jeff Schneider. Nonparametric divergence estimators for independent subspace analysis. In *European Signal Processing Conference (EUSIPCO)*, pages 1849–1853, 2011.

[69] Barnabás Póczos, Liang Xiong, and Jeff Schneider. Nonparametric divergence: Estimation with applications to machine learning on distributions. In *Uncertainty in Artificial Intelligence (UAI)*, pages 599–608, 2011.

[70] Ravikiran Rajagopal and Lee C. Potter. Multivariate MIMO FIR inverses. *IEEE Transactions on Image Processing*, 12:458–465, 2003.

[71] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method*. Springer, 2004.

[72] Alfréd Rényi. On measures of information and entropy. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1961.

[73] Marco Scarsini. On measures of concordance. *Stochastica*, 8:201–218, 1984.

[74] Friedrich Schmid, Rafael Schmidt, Thomas Blumentritt, Sandra Gaißer, and Martin Ruppert. *Copula Theory and Its Applications*, chapter Copula based Measures of Multivariate Association. Lecture Notes in Statistics. Springer, 2010.

[75] Tapio Schneider and Arnold Neumaier. Algorithm 808: ARfit - a Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software*, 27(1):58–65, 2001.

[76] B. Schweizer and E. F. Wolff. On nonparametric measures of dependence for random variables. *The Annals of Statistics*, 9:879–885, 1981.

[77] Dino Sejdinovic, Arthur Gretton, Bharath Sriperumbudur, and Kenji Fukumizu. Hypothesis testing using pairwise distances and associated kernels. In *International Conference on Machine Learning (ICML)*, pages 1111–1118, 2012.

[78] Sohan Seth and José C. Príncipe. On speeding up computation in information theoretic learning. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2883–2887, 2009.

[79] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[80] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[81] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences*, 23:301–321, 2003.

[82] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15:72–101, 1904.

[83] Kumar Sricharan and Alfred. O. Hero. Weighted k-NN graphs for Rényi entropy estimation in high dimensions. In *IEEE Workshop on Statistical Signal Processing (SSP)*, pages 773–776, 2011.

[84] Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

[85] Zoltán Szabó. Complete blind subspace deconvolution. In *International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 138–145, 2009.

[86] Zoltán Szabó. Autoregressive independent process analysis with missing observations. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 159–164, 2010.

[87] Zoltán Szabó and András Lőrincz. Towards independent subspace analysis in controlled dynamical systems. In *ICA Research Network International Workshop (ICARN)*, pages 9–12, 2008.

[88] Zoltán Szabó and András Lőrincz. Complex independent process analysis. *Acta Cybernetica*, 19:177–190, 2009.

[89] Zoltán Szabó and András Lőrincz. Fast parallel estimation of high dimensional information theoretical quantities with low dimensional random projection ensembles. In *International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 146–153, 2009.

[90] Zoltán Szabó and András Lőrincz. Distributed high dimensional information theoretical image registration via random projections. *Digital Signal Processing*, 22(6):894–902, 2012.

[91] Zoltán Szabó and Barnabás Póczos. Nonparametric independent process analysis. In *European Signal Processing Conference (EUSIPCO)*, pages 1718–1722, 2011.

[92] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Cross-entropy optimization for independent process analysis. In *International Conference on Independent Component Analysis and Blind Source Separation (ICA)*, pages 909–916, 2006.

[93] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Undercomplete blind subspace deconvolution. *Journal of Machine Learning Research*, 8:1063–1095, 2007.

[94] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Undercomplete blind subspace deconvolution via linear prediction. In *European Conference on Machine Learning (ECML)*, pages 740–747, 2007.

[95] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Auto-regressive independent process analysis without combinatorial efforts. *Pattern Analysis and Applications*, 13:1–13, 2010.

[96] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Separation theorem for independent subspace analysis and its consequences. *Pattern Recognition*, 45:1782–1791, 2012.

[97] Zoltán Szabó, Barnabás Póczos, Gábor Szirtes, and András Lőrincz. Post nonlinear independent subspace analysis. In *International Conference on Artificial Neural Networks (ICANN)*, pages 677–686, 2007.

[98] Zoltán Szabó and András Lőrincz. Real and complex independent subspace analysis by generalized variance. In *ICA Research Network International Workshop (ICARN)*, pages 85–88, 2006.

[99] Gábor J. Székely and Maria L. Rizzo and. Brownian distance covariance. *The Annals of Applied Statistics*, 3:1236–1265, 2009.

[100] Gábor J. Székely and Maria L. Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5, 2004.

[101] Gábor J. Székely and Maria L. Rizzo. A new test for multivariate normality. *Journal of Multivariate Analysis*, 93:58–80, 2005.

[102] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35:2769–2794, 2007.

[103] Anisse Taleb and Christian Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 10(47):2807–2820, 1999.

[104] Fabian J. Theis. Blind signal separation into groups of dependent signals using joint block diagonalization. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 5878–5881, 2005.

[105] Fabian J. Theis. Towards a general independent subspace analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1361–1368, 2007.

[106] Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479–487, 1988.

[107] James V. Uspensky. Asymptotic formulae for numerical functions which occur in the theory of partitions. *Bulletin of the Russian Academy of Sciences*, 14(6):199–218, 1920.

[108] Oldrich Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society, Series B*, 38:54–59, 1976.

[109] T. Villmann and S. Haase. Mathematical aspects of divergence based vector quantization using Fréchet-derivatives. Technical report, University of Applied Sciences Mittweida, 2010.

[110] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.

[111] Mark P. Wachowiak, Renata Smolikova, Georgia D. Tourassi, and Adel S. Elmaghraby. Estimation of generalized entropies with sample spacing. *Pattern Analysis and Applications*, 8:95–101, 2005.

[112] Quing Wang, Sanjeev R. Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55:2392–2405, 2009.

[113] Quing Wang, Sanjeev R. Kulkarni, and Sergio Verdú. Universal estimation of information measures for analog sources. *Foundations And Trends In Communications And Information Theory*, 5:265–353, 2009.

[114] Donghui Yan, Ling Huang, and Michael I. Jordan. Fast approximate spectral clustering. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 907–916, 2009.

[115] Joseph E. Yukich. Probability theory of classical Euclidean optimization problems. *Lecture Notes in Mathematics*, 1675, 1998.

[116] Andreas Ziehe, Motoaki Kawanabe, Stefan Harmeling, and Klaus-Robert Müller. Blind separation of postnonlinear mixtures using linearizing transformations and temporal decorrelation. *Journal of Machine Learning Research*, 4:1319–1338, 2003.

[117] V. M. Zolotarev. Probability metrics. *Theory of Probability and its Applications*, 28:278–302, 1983.