Information Theoretical Estimators (ITE) Toolbox
Release 0.63

Zoltán Szabó

June 9, 2016

Contents

List of Figures

List of Tables

List of Examples

List of Templates

1 Introduction

Since the pioneering work of Shannon [143], *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 tools to define the 'distance' of probability measures. Particularly, in the classical Shannon case, these three concepts can be naturally 'ordered': 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 [25]. 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 [10, 190, 186, 9, 115].

Independent component analysis (ICA) [69, 20, 22] 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 recordings (observations). 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 [18, 30]. 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 [18]: 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 [167][1], 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 [170].

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[2]. 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 focuses on *continuous* variables and

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

   - **entropy**: Shannon entropy, Rényi entropy, Tsallis entropy (Havrda and Charvát entropy), complex entropy, Φ-entropy (f-entropy), Sharma-Mittal entropy,
   - **mutual information**: generalized variance (GV), kernel canonical correlation analysis (KCCA), kernel generalized variance (KGV), Hilbert-Schmidt independence criterion (HSIC), Shannon mutual information (total correlation, multi-information), L_2 mutual information, Rényi mutual information, Tsallis mutual information, copula-based kernel dependency, multivariate version of Hoeffding's Φ, Schweizer-Wolff's σ and κ, complex mutual information, Cauchy-Schwartz quadratic mutual information (QMI), Euclidean distance based QMI, distance covariance, distance correlation, approximate correntropy independence measure, χ^2 mutual information (Hilbert-Schmidt norm of the normalized cross-covariance operator, squared-loss mutual information, mean square contingency), Lancaster three-variable interaction,
   - **divergence**: Kullback-Leibler divergence (relative entropy, I directed divergence), L_2 divergence, Rényi divergence, Tsallis divergence, Hellinger distance, Bhattacharyya distance, maximum mean discrepancy (MMD, kernel distance, current distance), J-distance (symmetrised Kullback-Leibler divergence, J divergence), Cauchy-Schwartz divergence, Euclidean distance based divergence, energy distance (specifically the Cramer-Von Mises distance), Jensen-Shannon divergence, Jensen-Rényi divergence, K divergence, L divergence, f-divergence

[1] Note: an alternative, exciting proof idea for deflation type methods has recently appeared in [111].

[2] See for example, http://www.cs.man.ac.uk/~pococka4/MIToolbox.html, http://www.cs.tut.fi/~timhome/tim/tim.htm, http://cran.r-project.org/web/packages/infotheo or http://cran.r-project.org/web/packages/entropy/.

(Csiszár-Morimoto divergence, Ali-Silvey distance), non-symmetric Bregman distance (Bregman divergence), Jensen-Tsallis divergence, symmetric Bregman distance, Pearson χ^2 divergence (χ^2 distance), Sharma-Mittal divergence,

- **association measures**: multivariate extensions of Spearman's ρ (Spearman's rank correlation coefficient, grade correlation coefficient), correntropy, centered correntropy, correntropy coefficient, correntropy induced metric, centered correntropy induced metric, multivariate extension of Blomqvist's β (medial correlation coefficient), multivariate conditional version of Spearman's ρ, lower/upper tail dependence via conditional Spearman's ρ,
- **cross quantities**: cross-entropy,
- **kernels on distributions**: expected kernel (summation kernel, mean map kernel, set kernel, multi-instance kernel, ensemble kernel; special convolution kernel [54]), Bhattacharyya kernel (Bhattacharyya coefficient, Hellinger affinity), probability product kernel, Jensen-Shannon kernel, exponentiated Jensen-Shannon kernel, Jensen-Tsallis kernel, exponentiated Jensen-Rényi kernel(s), exponentiated Jensen-Tsallis kernel(s),
- **conditional entropy**: conditional Shannon entropy,
- **conditional mutual information**: conditional Shannon mutual information,

based on

- nonparametric methods[3]: k-nearest neighbors, generalized k-nearest neighbors, weighted k-nearest neighbors, minimum spanning trees, random projection, kernel techniques, ensemble methods, sample spacing, von Mises expansion,
- kernel density estimation (KDE), adaptive partitioning, maximum entropy distribution: 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-, (iv) complex valued-, (v) partially observed models, (vi) as well as systems with nonparametric source dynamics; which can be used in combinations as well.
4. with *quick tests* for studying the consistency/efficiency of the estimators: (i) analytical vs. estimated quantity, (ii) positive semi-definiteness of the Gram matrix associated to a distribution kernel, (iii) image registration, (iv) distribution regression (supervised entropy learning, aerosol prediction based on multispectral satellite images).

Technical details:

- **Author**: Zoltán Szabó (http://www.gatsby.ucl.ac.uk/~szabo/, zoltan.szabo@gatsby.ucl.ac.uk).
- **Citing**: If you use the ITE toolbox in your work, please cite the paper(s) [158] ([170]), see the .bib in Appendix A.
- **Homepage of the ITE toolbox**: https://bitbucket.org/szzoli/ite/. Comments, feedbacks are welcome.
- **Share your ITE application**: https://bitbucket.org/szzoli/ite/wiki.
- **ITE mailing list**: https://groups.google.com/d/forum/itetoolbox.
- **Follow ITE**: on Bitbucket (https://bitbucket.org/szzoli/ite/follow), Twitter (https://twitter.com/ITEtoolbox).

[3]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.

- **License**: GNU GPLv3 or later.
- **Platforms**: ITE 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.
- **Programming environments**: Matlab[4], Octave[5].
- **Software requirements**: The ITE toolkit is self-contained, it only needs
  - a Matlab or an Octave environment with standard toolboxes:
    - Matlab: Image Processing, Optimization, Statistics.
    - Octave[6]: 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 [such as gcc (Linux), Microsoft Visual C++ (Windows)] to (optionally) speed up certain computations.
- **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.

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 unconditional information theoretical quantities (entropy, mutual information, divergence, association and cross measures, kernels on distributions) and their realizations in ITE; Section 4 is about conditional quantities. In Section 5, 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. Beyond IPA, ITE provides quick tests to study the consistency/efficiency of the estimators (Section 6). Section 7 is about the organization of the directories of the ITE toolbox.
- Appendix: Citing information of the ITE package is provided in Appendix A. Abbreviations of the paper are listed in Appendix B (Table 34). Functions with Octave-specific adaptations are summarized in Appendix C (Table 35). Some further formal definitions (concordance ordering, measure of concordance and -dependence, semimetric space of negative type, (covariant) Hilbertian metric) are given in Appendix D to make the documentation self-contained. A brief summary (lookup table) of the computations related to entropy, mutual information, divergence, association and cross measures, and kernels on distributions can be found in Appendix E. Derivation of explicit expressions for information theoretical quantities (without reference in Section 6) can be found in Section F.

2 Installation and Embedded Packages

This section is about (i) the installation of the ITE toolbox (Section 2.1), and (ii) the embedded dedicated solvers (Section 2.2).

2.1 Installation

You can install ITE by the `ITE_install.m` script: cd to the 'code' directory in Matlab/Octave and run `ITE_install`. Running the script from Matlab/Octave, it

1. adds the main ITE directory with subfolders to the Matlab/Octave PATH,
2. downloads and extracts the ARfit package, and (iii) compiles the embedded C/C++ accelerations.[7]

[4] http://www.mathworks.com/products/matlab/
[5] http://www.gnu.org/software/octave/
[6] See http://octave.sourceforge.net/packages.php.
[7] ITE also offers purely Matlab/Octave implementations for the computation of Hoeffding's Φ, Edgeworth expansion based entropy approximation and CDSS. Without compilation, these Matlab/Octave implementations are evoked.

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; %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.
AOD:MISR1 dataset: downloading, extraction: started.
AOD:MISR1 dataset: downloading, extraction: ready.
ITE code 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.
KDP (kdpee.c, kdpeemex.c) compilation: started.
KDP (kdpee.c, kdpeemex.c) compilation: ready.
Adaptive partitioning based MI (mutin.cpp) compilation: started.
Adaptive partitioning based MI (mutin.cpp) compilation: ready.
-------------------
Installation tests:
ANN quick installation test: successful.
NCut quick installation test: successful.
ARfit quick installation test: successful.
knn quick installation test: successful.
KDP quick installation test: successful.
Adaptive partitioning based MI quick installation test: successful.
```

Notes:

- The `ITE_install.m` script automatically detects the working environment (Matlab/Octave) and performs the installation accordingly, for example, it deletes the ann wrapper not suitable for the current working environment.
- There are two additional functions, `ITE_add_to_path.m` and `ITE_remove_from_path.m` to add/remove the ITE code directory to/from the Matlab/Octave PATH. They work similarly to `ITE_install.m`, you cd to the 'code' directory, and run them.

Example 2 (Add ITE to/remove it from the PATH)

```
>>ITE_add_to_path; %after cd-ing to the code directory;
                  %goal: to start a new session with ITE if it is not on the PATH
                  %assumption: ITE has been previously installed (see ITE_install)
>>...
>>ITE_remove_from_path; %after cd-ing to the code directory;
                       %goal: remove ITE from the PATH at the end of the session
```

2.2 Embedded Packages

The purpose of embedding packages was 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 5.1)),
- 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, in a maximally Octave compatible way. 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 35). The functionalities extended by the external packages are also available in both environments (Table 1). A short summary of the embedded/downloaded packages (directory 'shared/embedded', 'shared/downloaded') is given below:

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[8].
   - **Note**: fast nearest neighbor alternative of `knnsearch` $\in$ Matlab: Statistics Toolbox.
4. **FastKICA** (directory 'shared/embedded/FastKICA', version 1.0):
   - **URL**: http://www.gatsby.ucl.ac.uk/~gretton/fastKicaFiles/fastkica.htm, http://www.gatsby.ucl.ac.uk/~gretton/fastKicaFiles/fastKICA.zip

[8] 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 $\leq$ 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.

- **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.

5. **NCut** (Normalized Cut, directory 'shared/embedded/NCut'; version 9):
   - **URL**: http://www.cis.upenn.edu/~jshi/software/, http://www.cis.upenn.edu/~jshi/software/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 '10) = spectral clustering'.

6. **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`.
     - The sqdistance function can give some *small, but negative* values on the diagonal of '`sqdistance(Y)`'; we correct this issue in the embedded function.

7. **TCA** (directory 'shared/embedded/TCA'; version 1.0):
   - **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 '6) = sqdistance'.

8. **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.

9. **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[9].
- **Environment**: Matlab, Octave.
- **Note**: alternative of '12) = ARfit' in AR identification.

10. **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 '5)=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.

11. **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.

12. **ARfit** (directory 'shared/downloaded/ARfit', version 'March 20, 2011')
    - **URL**: http://www.mathworks.com/matlabcentral/fileexchange/174-arfit, 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 '9) = E4' in AR identification.

13. **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.

14. **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.

[9] 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.

- **Environment**: Matlab, Octave.
- **Note**: Alternative of '3)=ANN' in finding k-nearest neighbors.

15. **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 σ and κ 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`').
16. **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**: KDE based estimation of Cauchy-Schwartz quadratic mutual information, Euclidean distance based quadratic mutual information; and associated divergences; correntropy, centered correntropy, correntropy coefficient.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
17. **KDP** (directory 'shared/embedded/KDP'; version '1.1.1')
    - **URL**: https://github.com/danstowell/kdpee
    - **License**: GNU GPLv3 or later.
    - **Solver**: adaptive partitioning based Shannon entropy estimation.
    - **Installation**: Run the included `mexme` command from the `mat_oct` subfolder. Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
    - **Note**: KDP (`mexme.m`) contains different compilation options on Linux/UNIX.
18. **PSD** (directory 'shared/embedded/PSD_SzegoT'):
    - **URL (author's homepage)**: http://sst.uni-paderborn.de/team/david-ramirez/.
    - **License**: GNU GPLv3 or later.
    - **Solver**: PSD (power spectral density) and Szegő's theorem based Shannon entropy and Kullback-Leibler divergence estimator.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
19. **combn** (directory 'shared/embedded/combn'):
    - **URL**: http://www.mathworks.co.uk/matlabcentral/fileexchange/7147-combn-4-3.
    - **License**: 2-clause BSD.
    - **Solver**: generates all combinations of n elements.
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
20. **MISR1** (directory 'shared/downloaded/MISR1')
    - **URL**: http://www.dabi.temple.edu/~vucetic/MIR.html, http://www.dabi.temple.edu/~vucetic/data/MIR_datasets.zip.
    - **Dataset**: aerosol optical depth prediction dataset based on satellite images [192].
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.
21. **MI_AP** (directory 'shared/embedded/combn')
    - **URL**: http://si.utia.cas.cz/downloadPT.htm.

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]
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 σ and κ	SWICA	Matlab, C++	Matlab, Octave	shared/embedded/SWICA
KDE based estimation[c]	ITL	Matlab	Matlab, Octave	shared/embedded/ITL
adaptive (k-d) partitioning	kdpee	Matlab, C	Matlab, Octave	shared/embedded/KDP
PSD representation based estimators	PSD	Matlab	Matlab, Octave	shared/embedded/PSD_SzegoT
all combinations of n elements	combn	Matlab	Matlab, Octave	shared/embedded/combn
aerosol prediction dataset	-	-	Matlab, Octave	shared/downloaded/MISR1
adaptive partitioning based MI	MI_AP	Matlab, C++	Matlab, Octave	shared/embedded/MI_AP
von Mises expansion based estimation	if-estimators	Matlab	Matlab, Octave	shared/embedded/if-estimators

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 8.
[c] KDE based estimation of Cauchy-Schwartz quadratic mutual information, Euclidean distance based quadratic mutual information; and associated divergences; correntropy, centered correntropy, correntropy coefficient.

- **License**: GNU GPLv3 or later.
- **Solver**: Adaptive partitioning based mutual information estimation [29, 28].
- **Installation**: Add it with subfolders to your Matlab/Octave PATH.
- **Environment**: Matlab, Octave.
- **Note**: One can also compile the included `mutin.cpp` to further accelerate computations (see '`mex mutin.cpp`').

22. **if-estimators** (directory 'shared/embedded/if-estimators')
    - **URL**: `https://github.com/kirthevasank/if-estimators` (commit d92b1e1 Apr 16, 2016).
    - **License**: GNU GPLv3 or later.
    - **Solver**: von Mises expansion (relying on influence functions) based estimators [72].
    - **Installation**: Add it with subfolders to your Matlab/Octave PATH.
    - **Environment**: Matlab, Octave.

3 Estimation of Unconditional Quantities

In this section we focus on the estimation of unconditional information theoretical quantities.[10] Particularly, the underlying idea how the estimators are implemented in ITE is 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, kernels on distributions:

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 via standard information theoretical

[10] For conditional quantities see Section 4.

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 [25], 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).

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

1. construct new estimators from existing ones, and
2. transparently use *any* of these estimators in information theoretical optimization problems (see Section 5) – provided that they follow a simple template described in Section 3.3. User-defined parameter values of the estimators are also detailed in the section.

Some general guidelines concerning the choice of estimators are the topic of Section 3.4. Section 6 is about the quick test of the estimators.

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, Section 3.1.6 we consider the estimation of mutual information, divergence, association-, cross measures and kernels on distributions, respectively.

3.1.1 Entropy Estimators

Let us start with a simple example: let our goal is to estimate the **Shannon entropy** [143]

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

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 $\mathbf{y}$; shortly $\mathbf{y}\sim f$.[11] 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 $\mathbf{y}$. In ITE the estimation can be carried out as simply as follows:

Example 3 (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
```

Note (on the importance and usage of *mult*): if the information theoretical quantity of interest can be estimated exactly with the same computational complexity as 'up to multiplicative constant' (i.e., up to '*proportionality*'; $mult=0$), then (i) the two computations ($mult=0$ or 1) are handled jointly in the implementation, and (ii) the *exact* quantity ($mult=1$) is computed. In this case we also follow the template structure of the estimators (see Section 3.3); this makes the uniform usage of estimators possible.

Alternative entropy measures of interest covered include

[11] Here, and in the sequel log denotes natural logarithm, i.e., the unit of the information theoretical measures is nat.

1. **Rényi entropy** [130]: 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) [181, 55]: is closely related to the Rényi entropy. It is 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:

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

3. **Φ-entropy (f-entropy[12]) [184]**: The Φ-entropy with weight w is defined as

$$H_{\Phi,w}(y) = \int_{\mathbb{R}} f(u)\Phi(f(u))w(u)\mathrm{d}u, \tag{7}$$

where $\mathbb{R} \ni y \sim f$.

4. **Sharma-Mittal entropy**: it is defined [144, 2] as

$$H_{\mathrm{SM},\alpha,\beta}(\mathbf{y}) = \frac{1}{1-\beta}\left[\left(\int_{\mathbb{R}^d} f^\alpha(\mathbf{u})\mathrm{d}\mathbf{u}\right)^{\frac{1-\beta}{1-\alpha}} - 1\right] \quad (\alpha > 0, \alpha \neq 1, \beta \neq 1). \tag{8}$$

The Rényi-, Tsallis- and Shannon- entropies are special cases of this family, in limit sense:

$$\lim_{\beta\to 1} H_{\mathrm{SM},\alpha,\beta} = H_{\mathrm{R},\alpha}, \qquad H_{\mathrm{SM},\alpha,\beta} = H_{\mathrm{T},\alpha} \quad (\alpha = \beta), \qquad \lim_{(\alpha,\beta)\to(1,1)} H_{\mathrm{SM},\alpha,\beta} = H. \tag{9}$$

Other entropies can be estimated similarly to the Shannon entropy H (see Example 3), for example

Example 4 (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 [75] ($S = \{1\}$), [147, 45] $S = \{k\}$; in ITE '`Shannon_kNN_k`') and $H_{\mathrm{R},\alpha}$ estimation methods [196, 81] ($S = \{k\}$; '`Renyi_kNN_k`'), ITE also provides 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`') [117], (ii) generalized nearest neighbor graphs ($S \subseteq \{1,\ldots,k\}$; '`Renyi_kNN_S`') [110], (iii) weighted k-nearest neighbors ('`Renyi_weightedkNN`') [151], and (iv) minimum spanning trees ('`Renyi_MST`') [196] 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`') [81]. Multivariate Edgeworth expansion [60], Voronoi region [91], k-d partitioning [153] and von Mises expansion [72] based Shannon entropy estimators are also available in ITE ('`Shannon_Edgeworth`', '`Shannon_Voronoi`', '`Shannon_KDP`', '`Shannon_vME`'). The Sharma-Mittal entropy can be estimated via k-nearest neighbors ('`SharmaM_kNN_k`'), or in an exponential family [16] with maximum likelihood estimation (MLE, '`SharmaM_expF`') and analytical expressions [103]. Estimators based on the latter principle are also available for the (special) Shannon, Rényi and the Tsallis entropy ('`Shannon_expF`', '`Renyi_expF`', '`Tsallis_expF`') [102, 103].

For the one-dimensional case ($d = 1$), beside the previous techniques, ITE offers

[12] Since f also denotes the density, we refer to the quantity as the Φ-entropy.

- sample spacing based estimators:
  - Shannon entropy: by approximating the slope of the inverse distribution function [185] ('`Shannon_spacing_V`') and its bias corrected variant [184] ('`Shannon_spacing_Vb`'). The method described in [23] applies locally linear regression ('`Shannon_spacing_LL`'). Piecewise constant/linear correction has been applied in [106] ('`Shannon_spacing_Vpconst`')/[34] ('`Shannon_spacing_Vplin`', '`Shannon_spacing_Vplin2`'). KDE based slope correction was performed at the endpoints in [107] ('`Shannon_spacing_VKDE`').
  - Rényi entropy: The idea of [185] and the empiric entropy estimator of order m has been recently generalized to Rényi entropies [188] ('`Renyi_spacing_V`', '`Renyi_spacing_E`'). A continuously differentiable sample spacing (CDSS) based quadratic Rényi entropy estimator was presented in [108] ('`qRenyi_CDSS`').
  - Φ-entropy: see [184] ('`Phi_spacing`').
- maximum entropy distribution based estimators for the Shannon entropy [25, 61] with different function sets, see '`Shannon_MaxEnt1`', '`Shannon_MaxEnt2`'.
- power spectral density and Szegő's theorem based estimation [47, 46, 127], see '`Shannon_PSD_SzegoT`'.

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 3 and Example 4, one only has to change '`Shannon_kNN_k`' (see Example 3) and '`Renyi_kNN_k`' (see Example 4) to the `cost_name` given in the last column of the table.

Note: the `Renyi_MST` method (see Table 2) estimates the H_α Rényi entropy up to an additive constant which depends on the dimension d and α, 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_MST`, the estimation instructions are as follows:

1. Set `co.alpha` (α) and `co.k` (k) in '`HRenyi_MST_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_MST_initialization.m`': '`co.additive_constant_is_relevant = 1`'.
4. Estimate the Rényi entropy (after initialization): '`HRenyi_MST_estimation.m`'.

3.1.2 Mutual Information Estimators

In our next example, we consider the estimation of the **mutual information**[13] of the d_m-dimensional components of the random variable $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right] \in \mathbb{R}^d$ ($d = \sum_{m=1}^M d_m$):

$$I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = \int_{\mathbb{R}^{d_1}} \cdots \int_{\mathbb{R}^{d_M}} f\left(\mathbf{u}^1, \ldots, \mathbf{u}^M\right) \log \left[\frac{f\left(\mathbf{u}^1, \ldots, \mathbf{u}^M\right)}{\prod_{m=1}^M f_m(\mathbf{u}^m)}\right] \mathrm{d}\mathbf{u}^1 \cdots \mathrm{d}\mathbf{u}^M \tag{10}$$

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 [25]. 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. An alternative is to use von Mises expansion [72] ('`Shannon_vME`' in ITE) or adaptive partitioning to directly estimate mutual information [29, 28] ('`Shannon_AP`', '`Shannon_AP2`' in ITE).

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

$$I_{\mathrm{KCCA}}(\mathbf{y}^1, \mathbf{y}^2) = \sup_{g_1 \in \mathcal{F}^1, g_2 \in \mathcal{F}^2} \frac{\mathrm{cov}[g_1(\mathbf{y}^1), g_2(\mathbf{y}^2)]}{\sqrt{\mathrm{var}\left[g_1(\mathbf{y}^1)\right] + \kappa \left\|g_1\right\|_{\mathcal{F}^1}^2} \sqrt{\mathrm{var}\left[g_2(\mathbf{y}^2)\right] + \kappa \left\|g_2\right\|_{\mathcal{F}^2}^2}}, \quad (\kappa > 0) \tag{11}$$

[13] Mutual information is also known in the literature as the special case of total correlation [193] or multi-information [154] when the number of subspaces is $M = 2$.

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'`
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)	`'Shannon_spacing_V'` with piecewise linear correction-2	$d = 1$	`'Shannon_spacing_Vplin2'`
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'`
Shannon entropy (H)	adaptive (k-d) partitioning, plug-in	$d \geq 1$	`'Shannon_KDP'`
Shannon entropy (H)	maximum entropy distribution, function set1, plug-in	$d = 1$	`'Shannon_MaxEnt1'`
Shannon entropy (H)	maximum entropy distribution, function set2, plug-in	$d = 1$	`'Shannon_MaxEnt2'`
Φ-entropy ($H_{\Phi,w}$)	sample spacing	$d = 1$	`'Phi_spacing'`
Sharma-Mittal entropy ($H_{\mathrm{SM},\alpha,\beta}$)	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'SharmaM_kNN_k'`
Sharma-Mittal entropy ($H_{\mathrm{SM},\alpha,\beta}$)	MLE + analytical value in the exponential family	$d \geq 1$	`'SharmaM_expF'`
Shannon entropy (H)	power spectral density, Szegő's theorem	$d = 1$	`'Shannon_PSD_SzegoT'`
Rényi entropy ($H_{\mathrm{R},\alpha}$)	MLE + analytical value in the exponential family	$d \geq 1$	`'Renyi_expF'`
Tsallis entropy ($H_{\mathrm{T},\alpha}$)	MLE + analytical value in the exponential family	$d \geq 1$	`'Tsallis_expF'`
Shannon entropy (H)	MLE + analytical value in the exponential family	$d \geq 1$	`'Shannon_expF'`
Shannon entropy (H)	`'Shannon_spacing_V'` with KDE based correction	$d = 1$	`'Shannon_spacing_VKDE'`
Shannon entropy (H)	von Mises expansion	$d \geq 1$	`'Shannon_vME'`

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

for $M = 2$ components, where 'cov' denotes covariance and 'var' stands for variance. In words, I_{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_{KCCA} can be reduced to a generalized eigenvalue problem and the measure can be extended to $M \geq 2$ components to measure pairwise independence [7, 167]. 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 [25]:

$$I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = -\frac{1}{2} \log \left(\frac{\det \mathbf{C}}{\prod_{m=1}^{M} \det \mathbf{C}^{m,m}} \right). \tag{12}$$

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

$$I_{\text{KGV}}\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = -\frac{1}{2} \log \left[\frac{\det(\boldsymbol{\mathcal{K}})}{\prod_{m=1}^{M} \det(\boldsymbol{\mathcal{K}}^{m,m})} \right], \tag{13}$$

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 [7, 167]. 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

$$\mathbf{C}_{\mathbf{y}^1,\mathbf{y}^2} = \mathbb{E}\left(\left[\boldsymbol{\varphi}_1(\mathbf{y}^1) - \boldsymbol{\mu}_1\right] \otimes \left[\boldsymbol{\varphi}_2(\mathbf{y}^2) - \boldsymbol{\mu}_2\right]\right), \tag{14}$$

where $\otimes$ denotes tensor product, $\mathbb{E}$ is the expectation and the mean embeddings [11] are

$$\boldsymbol{\mu}_m = \mathbb{E}[\boldsymbol{\varphi}_m(\mathbf{y}^m)] \quad (m = 1, 2). \tag{15}$$

HSIC [51] is defined as the Hilbert-Schmidt norm of the cross-covariance operator

$$I_{\text{HSIC}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \left\|\mathbf{C}_{\mathbf{y}^1,\mathbf{y}^2}\right\|_{\text{HS}}^2. \tag{16}$$

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

$$\begin{aligned}\left[I_{\text{HSIC}}\left(\mathbf{y}^1, \mathbf{y}^2\right)\right]^2 = \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{1'},\mathbf{y}^{2'}} k_1\left(\mathbf{y}^1, \mathbf{y}^{1'}\right) k_2\left(\mathbf{y}^2, \mathbf{y}^{2'}\right) + \mathbb{E}_{\mathbf{y}^1}\mathbb{E}_{\mathbf{y}^{1'}} k_1\left(\mathbf{y}^1, \mathbf{y}^{1'}\right) \mathbb{E}_{\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{2'}} k_1\left(\mathbf{y}^2, \mathbf{y}^{2'}\right) \\ - 2\mathbb{E}_{\mathbf{y}^{1'}\mathbf{y}^{2'}}\left[\mathbb{E}_{\mathbf{y}^1} k_1\left(\mathbf{y}^1, \mathbf{y}^{1'}\right) \mathbb{E}_{\mathbf{y}^1} k_2\left(\mathbf{y}^2, \mathbf{y}^{2'}\right)\right],\end{aligned} \tag{17}$$

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 [160] 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 $\mathcal{F}$:

$$I_{\text{GV}}\left(y^1, y^2\right) = \sum_{g \in \mathcal{F}} \left(corr\left[g\left(y^1\right), g\left(y^2\right)\right]\right)^2. \tag{18}$$

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

5. **Hoeffding's Φ, Schweizer-Wolff's σ and κ**: 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 $(F_i,\ i = 1, \ldots, d)$ [148]:

$$F(\mathbf{y}) = C\left(F_1\left(y^1\right), \ldots, F_d\left(y^d\right)\right), \tag{19}$$

or in other words

$$C(\mathbf{u}) = \mathbb{P}(\mathbf{U} \leq \mathbf{u}), \quad (\mathbf{u} = [u_1; \ldots; u_d] \in [0,1]^d), \tag{20}$$

where

$$\mathbf{U} = \left[F_1\left(y^1\right); \ldots; F_d\left(y^d\right)\right] \in [0,1]^d. \tag{21}$$

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 Π defined as

$$\Pi(u_1, \ldots, u_d) = \prod_{i=1}^{d} u_i. \tag{22}$$

Using this result, the independence of y^is can be measured by the (normalized) L^p distance of C and Π:

$$\left(h_p(d) \int_{[0,1]^d} |C(\mathbf{u}) - \Pi(\mathbf{u})|^p \, \mathrm{d}\mathbf{u}\right)^{\frac{1}{p}}, \tag{23}$$

where (i) $1 \leq p \leq \infty$, and (ii) by an appropriate choice of the normalization constant $h_p(d)$, the value of (23) belongs to the interval $[0,1]$ for any C.

- For $p = 2$, the special

$$I_\Phi\left(y^1, \ldots, y^d\right) = I_\Phi(C) = \left(h_2(d) \int_{[0,1]^d} [C(\mathbf{u}) - \Pi(\mathbf{u})]^2 \mathrm{d}\mathbf{u}\right)^{\frac{1}{2}} \tag{24}$$

  quantity
  - is a generalization of Hoeffding's Φ defined for $d = 2$ [58],
  - whose empirical estimation can be analytically computed [42].

  The name of the objective is '`Hoeffding`' in ITE.

- For $p = 1$ and $p = \infty$, one obtains Schweizer-Wolff's σ and κ [138, 194]. The first measure (I_{SW1}) satisfies all the properties of a multivariate measure of dependence in the sense of Def. 5 (see Section D); the second index (I_{SWinf}) fullfills D1, D2, D5, D6 and D8 of Def. 5 [194].
  For $p \in \{1, \infty\}$ no explicit expressions for the integrals in Eq.(23) are available. For small dimensional problems, however, the quantities can be efficiently estimated numerically. ITE contains methods for the $M = 2$ case:

$$I_{\mathrm{SW1}}\left(y^1, y^2\right) = I_{\mathrm{SW1}}(C) = \sigma = 12 \int_{[0,1]^2} |C(\mathbf{u}) - \Pi(\mathbf{u})| \mathrm{d}\mathbf{u}, \tag{25}$$

$$I_{\mathrm{SWinf}}\left(y^1, y^2\right) = I_{\mathrm{SWinf}}(C) = \kappa = 4 \sup_{\mathbf{u} \in [0,1]^2} |C(\mathbf{u}) - \Pi(\mathbf{u})|. \tag{26}$$

  The two measures are called '`SW1`' and '`SWinf`'.

For an excellent introduction on copulas, see [98].

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 [142]:

$$I_{\text{QMI-CS}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \log\left[\frac{\left(\int_{\mathbb{R}^{d_1}} \int_{\mathbb{R}^{d_2}} \left[f\left(\mathbf{u}^1, \mathbf{u}^2\right)\right]^2 \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2\right)\left(\int_{\mathbb{R}^{d_1}} \int_{\mathbb{R}^{d_2}} \left[f_1\left(\mathbf{u}^1\right)\right]^2 \left[f_2\left(\mathbf{u}^2\right)\right]^2 \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2\right)}{\left[\int_{\mathbb{R}^{d_1}} \int_{\mathbb{R}^{d_2}} f\left(\mathbf{u}^1, \mathbf{u}^2\right) f_1\left(\mathbf{u}^1\right) f_2\left(\mathbf{u}^2\right) \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2\right]^2}\right], \tag{27}$$

$$\begin{aligned} I_{\text{QMI-ED}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = &\left(\int_{\mathbb{R}^{d_1}} \int_{\mathbb{R}^{d_2}} \left[f\left(\mathbf{u}^1, \mathbf{u}^2\right)\right]^2 \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2\right) + \left(\int_{\mathbb{R}^{d_1}} \int_{\mathbb{R}^{d_2}} \left[f_1\left(\mathbf{u}^1\right)\right]^2 \left[f_2\left(\mathbf{u}^2\right)\right]^2 \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2\right) \\ &- 2 \int_{\mathbb{R}^{d_1}} \int_{\mathbb{R}^{d_2}} f\left(\mathbf{u}^1, \mathbf{u}^2\right) f_1\left(\mathbf{u}^1\right) f_2\left(\mathbf{u}^2\right) \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2. \end{aligned} \tag{28}$$

The measures can

(a) be approximated in ITE via

$$\hat{f}_m(\mathbf{u}) = \frac{1}{T}\sum_{t=1}^{T} k\left(\mathbf{u} - \mathbf{y}_t^m\right) \tag{29}$$

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. (58), (59)]:

$$I_{\text{QMI-CS}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = D_{\text{CS}}\left(f, f_1 f_2\right), \tag{30}$$

$$I_{\text{QMI-ED}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = D_{\text{ED}}\left(f, f_1 f_2\right). \tag{31}$$

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 [175, 174]. 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 φ_j (φ_{12}) stand for the characteristic function of $\mathbf{y}^j$ ($\left[\mathbf{y}^1; \mathbf{y}^2\right]$):

$$\varphi_{12}\left(\mathbf{u}^1, \mathbf{u}^2\right) = \mathbb{E}\left[e^{i\left\langle \mathbf{u}^1, \mathbf{y}^1\right\rangle + i\left\langle \mathbf{u}^2, \mathbf{y}^2\right\rangle}\right], \tag{32}$$

$$\varphi_j\left(\mathbf{u}^j\right) = \mathbb{E}\left[e^{i\left\langle \mathbf{u}^j, \mathbf{y}^j\right\rangle}\right], \quad (j = 1, 2) \tag{33}$$

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

$$I_{\text{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \left\|\varphi_{12} - \varphi_1\varphi_2\right\|_{L_w^2} = \sqrt{\int_{\mathbb{R}^{d_1+d_2}} \left|\varphi_{12}\left(\mathbf{u}^1, \mathbf{u}^2\right) - \varphi_1\left(\mathbf{u}^1\right)\varphi_2\left(\mathbf{u}^2\right)\right|^2 w\left(\mathbf{u}^1, \mathbf{u}^2\right) \mathrm{d}\mathbf{u}^1 \mathrm{d}\mathbf{u}^2} \tag{34}$$

with a suitable chosen w weight function

$$w\left(\mathbf{u}^1, \mathbf{u}^2\right) = \frac{1}{c(d_1, \alpha)c(d_2, \alpha)\left[\left\|\mathbf{u}^1\right\|_2\right]^{d_1+\alpha}\left[\left\|\mathbf{u}^2\right\|_2\right]^{d_2+\alpha}}, \tag{35}$$

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

$$c(d, \alpha) = \frac{2\pi^{\frac{d}{2}}\Gamma\left(1 - \frac{\alpha}{2}\right)}{\alpha 2^{\alpha}\Gamma\left(\frac{d+\alpha}{2}\right)}. \tag{36}$$

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

$$I_{\text{dVar}}\left(\mathbf{y}^j, \mathbf{y}^j\right) = \left\|\varphi_{jj} - \varphi_j\varphi_j\right\|_{L_w^2}. \tag{37}$$

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

$$I_{\text{dCor}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = \begin{cases} \frac{I_{\text{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right)}{\sqrt{I_{\text{dVar}}\left(\mathbf{y}^1, \mathbf{y}^1\right) I_{\text{dVar}}\left(\mathbf{y}^2, \mathbf{y}^2\right)}}, & \text{if } I_{\text{dVar}}\left(\mathbf{y}^1, \mathbf{y}^1\right) I_{\text{dVar}}\left(\mathbf{y}^2, \mathbf{y}^2\right) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{38}$$

a type of unsigned correlation. By construction $I_{\text{dCor}}\left(\mathbf{y}^1, \mathbf{y}^2\right) \in [0, 1]$, 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.

8. **Lancaster three-variable interaction**: Let $\mathbb{P}$ denote the probability measure of $\mathbf{y} = \left[\mathbf{y}^1; \mathbf{y}^2; \mathbf{y}^3\right] \in \mathcal{Y}_1 \times \mathcal{Y}_2 \times \mathcal{Y}_3$ and $\mathbb{P}_S$ ($S \subseteq \{1, 2, 3\}$) the associated marginals; each $\mathcal{Y}_u$ space is assumed to be endowed with a kernels k_u ($u = 1, 2, 3$). The *Lancaster 3-variable interaction* [79], which is a signed measure, is

- defined as

$$L(\mathbb{P}) = \mathbb{P}_{1,2,3} - \mathbb{P}_{1,2}\mathbb{P}_3 - \mathbb{P}_{2,3}\mathbb{P}_1 - \mathbb{P}_{1,3}\mathbb{P}_2 + 2\mathbb{P}_1\mathbb{P}_2\mathbb{P}_3. \tag{39}$$

- guaranteed to be zero, if $\mathbb{P}$ can be factorised as a product of its (possible) multidimensional marginals

$$\left(\left[\mathbf{y}^1;\mathbf{y}^2\right] \perp\!\!\!\perp \mathbf{y}^3\right) \vee \left(\left[\mathbf{y}^1;\mathbf{y}^3\right] \perp\!\!\!\perp \mathbf{y}^2\right) \vee \left(\left[\mathbf{y}^2;\mathbf{y}^3\right] \perp\!\!\!\perp \mathbf{y}^1\right) \Rightarrow L(\mathbb{P}) = 0. \tag{40}$$

  Here, '$\perp\!\!\!\perp$' means independence and '$\vee$' denotes the logical 'or'. For example, '$\left[\mathbf{y}^1;\mathbf{y}^2\right] \perp\!\!\!\perp \mathbf{y}^3$' stands for $\mathbb{P} = \mathbb{P}_{1,2}\mathbb{P}_3$.

The Lancaster three-variable interaction based dependency index is defined [139] as the squared norm of the mean embedded (see also (15)) L

$$I_{\text{3-Lanc}}\left(\mathbf{y}^1,\mathbf{y}^2,\mathbf{y}^3\right) = \left\|\boldsymbol{\mu}_{L(\mathbb{P})}\right\|^2_{k_1\otimes k_2\otimes k_3}. \tag{41}$$

The estimator is called '`3way_Lancaster`' in ITE.

9. **Three-variable joint independence measure**: If one takes the mean embedding (see also (15)) of the signed measure

$$J(\mathbb{P}) = \mathbb{P}_{1,2,3} - \mathbb{P}_1\mathbb{P}_2\mathbb{P}_3 \tag{42}$$

instead of $L(\mathbb{P})$ [Eq. (39)], then a similar index can be defined [139]:

$$I_{\text{3-joint}}\left(\mathbf{y}^1,\mathbf{y}^2,\mathbf{y}^3\right) = \left\|\boldsymbol{\mu}_{J(\mathbb{P})}\right\|^2_{k_1\otimes k_2\otimes k_3}. \tag{43}$$

$I_{\text{3-joint}}$ measures the joint independence of its arguments; the associated estimator is called '`3way_joint`' 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 5 (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 identical: 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, or I directed divergence) [76][14]:

$$D(f_1, f_2) = \int_{\mathbb{R}^d} f_1(\mathbf{u}) \log\left[\frac{f_1(\mathbf{u})}{f_2(\mathbf{u})}\right] \mathrm{d}\mathbf{u}. \tag{44}$$

In practise, one has independent, i.i.d. samples from f_1 and f_2, $\{\mathbf{y}^1_t\}_{t=1}^{T_1}$ and $\{\mathbf{y}^2_t\}_{t=1}^{T_2}$, respectively. The goal is to estimate divergence D using these samples. Of course, there exist many variants/extensions of the traditional Kullback-Leibler divergence [186, 9]; 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**:

$$D_{\text{L}}(f_1, f_2) = \sqrt{\int_{\mathbb{R}^d} \left[f_1(\mathbf{u}) - f_2(\mathbf{u})\right]^2 \mathrm{d}\mathbf{u}}. \tag{45}$$

   By definition $D_{\text{L}}(f_1, f_2)$ is non-negative, and is zero if and only if $f_1 = f_2$.

[14] $D(f_1, f_2) \geq 0$ with equality iff $f_1 = f_2$. The Kullback-Leibler divergence is a special f-divergence (with $f(t) = t\log(t)$), see (143).

Estimated quantity	Principle	d_m	M	`cost_name`
generalized variance (I_{GV})	f-covariance/-correlation ($f \in \mathcal{F}$, $\lvert\mathcal{F}\rvert < \infty$)	$d_m = 1$	$M = 2$	`'GV'`
Hilbert-Schmidt indep. criterion (I_{HSIC})	HS norm of the cross-covariance operator	$d_m \geq 1$	$M \geq 2$	`'HSIC'`
kernel canonical correlation (I_{KCCA})	sup correlation over RKHSs	$d_m \geq 1$	$M \geq 2$	`'KCCA'`
kernel generalized variance (I_{KGV})	Gaussian mutual information of the features	$d_m \geq 1$	$M \geq 2$	`'KGV'`
Hoeffding's Φ (I_{Φ}), multivariate	L^2 distance of the joint- and the product copula	$d_m = 1$	$M \geq 2$	`'Hoeffding'`
Schweizer-Wolff's σ (I_{SW1})	L^1 distance of the joint- and the product copula	$d_m = 1$	$M = 2$	`'SW1'`
Schweizer-Wolff's κ (I_{SWinf})	L^∞ 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_{dCov})	pairwise distances	$d_m \geq 1$	$M = 2$	`'dCov'`
distance correlation (I_{dCor})	pairwise distances	$d_m \geq 1$	$M = 2$	`'dCor'`
Lancaster 3-variable interaction ($I_{\mathrm{3\text{-}Lanc}}$)	embedding of the Lancaster interaction measure	$d_m \geq 1$	$M = 3$	`'3way_Lancaster'`
3-variable joint independence ($I_{\mathrm{3\text{-}joint}}$)	embedding of the 'joint - product of marginals'	$d_m \geq 1$	$M = 3$	`'3way_joint'`
(Shannon) mutual information (I)	adaptive partitioning	$d_m = 1$	$M = 2$	`'Shannon_AP2'`
(Shannon) mutual information (I)	adaptive partitioning	$d_m = 1$	$M \geq 2$	`'Shannon_AP'`
(Shannon) mutual information (I)	von Mises expansion	$d_m \geq 1$	$M = 2$	`'Shannon_vME'`

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**:

$$D_{\mathrm{T},\alpha}(f_1, f_2) = \frac{1}{\alpha - 1}\left(\int_{\mathbb{R}^d} f_1^\alpha(\mathbf{u}) f_2^{1-\alpha}(\mathbf{u}) \mathrm{d}\mathbf{u} - 1\right) \quad (\alpha \in \mathbb{R} \setminus \{1\}). \tag{46}$$

Notes:

- The Kullback-Leibler divergence [Eq. (44)] is a special of Tsallis' in limit sense:

$$\lim_{\alpha \to 1} D_{\mathrm{T},\alpha} = D. \tag{47}$$

- As a function of α, the sign of the Tsallis divergence is as follows:

$$\alpha < 0 \Rightarrow D_{\mathrm{T},\alpha}(f_1, f_2) \leq 0, \qquad \alpha = 0 \Rightarrow D_{\mathrm{T},\alpha}(f_1, f_2) = 0, \qquad \alpha > 0 \Rightarrow D_{\mathrm{T},\alpha}(f_1, f_2) \geq 0. \tag{48}$$

3. **Rényi divergence**:

$$D_{\mathrm{R},\alpha}(f_1, f_2) = \frac{1}{\alpha - 1} \log \int_{\mathbb{R}^d} f_1^\alpha(\mathbf{u}) f_2^{1-\alpha}(\mathbf{u}) \mathrm{d}\mathbf{u} \quad (\alpha \in \mathbb{R} \setminus \{1\}). \tag{49}$$

Notes:

- The Kullback-Leibler divergence [Eq. (44)] is a special of Rényi's in limit sense:

$$\lim_{\alpha \to 1} D_{\mathrm{R},\alpha} = D. \tag{50}$$

- As a function of α, the sign of the Rényi divergence is as follows:

$$\alpha < 0 \Rightarrow D_{\mathrm{R},\alpha}(f_1, f_2) \leq 0, \qquad \alpha = 0 \Rightarrow D_{\mathrm{R},\alpha}(f_1, f_2) = 0, \qquad \alpha > 0 \Rightarrow D_{\mathrm{R},\alpha}(f_1, f_2) \geq 0. \tag{51}$$

4. **Maximum mean discrepancy** (MMD, also called the kernel distance and current distance) [49]:

$$D_{\mathrm{MMD}}(f_1, f_2) = \left\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right\|_{\mathcal{F}}, \tag{52}$$

where $\boldsymbol{\mu}_m$ is the mean embedding [11] of f_m ($m = 1, 2$) and $\mathcal{F} = \mathcal{F}^1 = \mathcal{F}^2$, see the definition of HSIC [Eq. (15)]. Notes:

- $D_{\mathrm{MMD}}(f_1, f_2)$ is a Hilbertian metric [56, 48]; see Def. 7.
- In the statistics literature, MMD is known as an integral probability metric (IPM) [199, 95, 152]:

$$D_{\mathrm{MMD}}(f_1, f_2) = \sup_{g \in \mathcal{B}} \left(\mathbb{E}\left[g\left(\mathbf{y}^1\right)\right] - \mathbb{E}[g\left(\mathbf{y}^2\right)] \right), \tag{53}$$

  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. (114)-(115)):

$$I_{\mathrm{HSIC}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = D_{\mathrm{MMD}}(f, f_1 f_2), \tag{54}$$

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

$$\left[D_{\mathrm{MMD}}(f_1, f_2)\right]^2 = \mathbb{E}_{\mathbf{y}^1, \mathbf{y}^{1'}}\left[k\left(\mathbf{y}^1, \mathbf{y}^{1'}\right)\right] + \mathbb{E}_{\mathbf{y}^2, \mathbf{y}^{2'}}\left[k\left(\mathbf{y}^2, \mathbf{y}^{2'}\right)\right] - 2\mathbb{E}_{\mathbf{y}^1, \mathbf{y}^2}\left[k\left(\mathbf{y}^1, \mathbf{y}^2\right)\right], \tag{55}$$

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

5. **Hellinger distance**:

$$D_{\mathrm{H}}(f_1, f_2) = \sqrt{\frac{1}{2}\int_{\mathbb{R}^d}\left[\sqrt{f_1(\mathbf{u})} - \sqrt{f_2(\mathbf{u})}\right]^2 \mathrm{d}\mathbf{u}} = \sqrt{1 - \int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}}. \tag{56}$$

   Notes:

   - As it is known $D_{\mathrm{H}}(f_1, f_2)$ is a (covariant) Hilbertian metric [56]; see Def. 7.
   - D_{H}^2 is a special f-divergence [with $f(t) = \frac{1}{2}(\sqrt{t} - 1)^2$], see (143).

6. **Bhattacharyya distance**:

$$D_{\mathrm{B}}(f_1, f_2) = -\log\left(\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}\right). \tag{57}$$

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

$$D_{\mathrm{CS}}\left(f_1, f_2\right) = \log\left[\frac{\left(\int_{\mathbb{R}^d}\left[f_1\left(\mathbf{u}\right)\right]^2 \mathrm{d}\mathbf{u}\right)\left(\int_{\mathbb{R}^d}\left[f_2\left(\mathbf{u}\right)\right]^2 \mathrm{d}\mathbf{u}\right)}{\left(\int_{\mathbb{R}^d} f_1(\mathbf{u}) f_2(\mathbf{u})\mathrm{d}\mathbf{u}\right)^2}\right] = \log\left[\frac{1}{\cos^2(f_1, f_2)}\right], \tag{58}$$

$$D_{\mathrm{ED}}\left(f_1, f_2\right) = \int_{\mathbb{R}^d}\left[f_1\left(\mathbf{u}\right)\right]^2 \mathrm{d}\mathbf{u} + \int_{\mathbb{R}^d}\left[f_2\left(\mathbf{u}\right)\right]^2 \mathrm{d}\mathbf{u} - 2\int_{\mathbb{R}^d} f_1(\mathbf{u}) f_2(\mathbf{u})\mathrm{d}\mathbf{u} = \int_{\mathbb{R}^d}\left[f_1(\mathbf{u}) - f_2(\mathbf{u})\right]^2 \mathrm{d}\mathbf{u} \tag{59}$$

$$= \left[D_{\mathrm{L}}(f_1, f_2)\right]^2. \tag{60}$$

8. **Energy distance**: Let $(\mathcal{Z}, \rho)$ be a semimetric space of negative type (see Def. 6, 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 [172, 173]:

$$D_{\mathrm{EnDist}}(f_1, f_2) = 2\mathbb{E}\left[\rho\left(\mathbf{y}^1, \mathbf{y}^2\right)\right] - \mathbb{E}\left[\rho\left(\mathbf{y}^1, \mathbf{y}^{1'}\right)\right] - \mathbb{E}\left[\rho\left(\mathbf{y}^2, \mathbf{y}^{2'}\right)\right]. \tag{61}$$

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

$$D_{\mathrm{EnDist}}(f_1, f_2) = 2\mathbb{E}\left\|\mathbf{y}^1 - \mathbf{y}^2\right\|_2 - \mathbb{E}\left\|\mathbf{y}^1 - \mathbf{y}^{1'}\right\|_2 - \mathbb{E}\left\|\mathbf{y}^2 - \mathbf{y}^{2'}\right\|_2. \tag{62}$$

   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`'.

9. **Bregman distance**: The non-symmetric Bregman distance (also called Bregman divergence) is defined [15, 27, 81] as

$$D_{\mathrm{NB},\alpha}(f_1,f_2)=\int_{\mathbb{R}^d}\left[f_2^\alpha(\mathbf{u})+\frac{1}{\alpha-1}f_1^\alpha(\mathbf{u})-\frac{\alpha}{\alpha-1}f_1(\mathbf{u})f_2^{\alpha-1}(\mathbf{u})\right]\mathrm{d}\mathbf{u},\quad(\alpha\neq 1). \tag{63}$$

10. **Symmetric Bregman distance**: The symmetric Bregman distance is defined [15, 27, 81] via its non-symmetric counterpart:

$$D_{\mathrm{SB},\alpha}(f_1,f_2)=\frac{1}{\alpha}\left[D_{\mathrm{NB},\alpha}(f_1,f_2)+D_{\mathrm{NB},\alpha}(f_2,f_1)\right],\quad(\alpha\neq 1) \tag{64}$$

$$=\frac{1}{\alpha-1}\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})-f_2(\mathbf{u})\right]\left[f_1^{\alpha-1}(\mathbf{u})-f_2^{\alpha-1}(\mathbf{u})\right]\mathrm{d}\mathbf{u} \tag{65}$$

$$=\frac{1}{\alpha-1}\int_{\mathbb{R}^d}f_1^\alpha(\mathbf{u})+f_2^\alpha(\mathbf{u})-f_1(\mathbf{u})f_2^{\alpha-1}(\mathbf{u})-f_2(\mathbf{u})f_1^{\alpha-1}(\mathbf{u})\mathrm{d}\mathbf{u}. \tag{66}$$

Specially, for $\alpha=2$ we obtain the square of the L_2-divergence [see Eq. (45)]:

$$\left[D_{\mathrm{L}}(f_1,f_2)\right]^2=D_{\mathrm{NB},2}(f_1,f_2)=D_{\mathrm{SB},2}(f_1,f_2). \tag{67}$$

11. **Pearson χ^2 divergence**: The Pearson χ^2 divergence (χ^2 distance) is defined [112] as

$$D_{\chi^2}(f_1,f_2)=\int_{supp(f_1)\cup supp(f_2)}\frac{\left[f_1(\mathbf{u})-f_2(\mathbf{u})\right]^2}{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}=\int_{supp(f_1)\cup supp(f_2)}\frac{\left[f_1(\mathbf{u})\right]^2}{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}-1, \tag{68}$$

where $supp(f_i)$ denotes the support of f_i.

Note: $D_{\chi^2}(f_1,f_2)\geq 0$, and is zero iff $f_1=f_2$. D_{χ^2} is a special f-divergence with $f(t)=(t-1)^2$, see (143).

12. **Sharma-Mittal divergence**: The Sharma-Mittal divergence [89] (see also (71)) is defined as

$$D_{\mathrm{SM},\alpha,\beta}(f_1,f_2)=\frac{1}{\beta-1}\left[\left(\int_{\mathbb{R}^d}[f_1(\mathbf{u})]^\alpha[f_2(\mathbf{u})]^{1-\alpha}\mathrm{d}\mathbf{u}\right)^{\frac{1-\beta}{1-\alpha}}-1\right]\quad(0<\alpha\neq 1,\beta\neq 1) \tag{69}$$

$$=\frac{1}{\beta-1}\left(\left[D_{\mathrm{temp1}}(\alpha)\right]^{\frac{1-\beta}{1-\alpha}}-1\right). \tag{70}$$

Note: It is known that $D_{\mathrm{SM},\alpha,\beta}(f_1,f_2)=0$, if and only if $f_1=f_2$.

Let us note that for (46), (49), (56) and (57), it is sufficient to estimate the

$$D_{\mathrm{temp1}}(\alpha)=\int_{\mathbb{R}^d}[f_1(\mathbf{u})]^\alpha[f_2(\mathbf{u})]^{1-\alpha}\mathrm{d}\mathbf{u} \tag{71}$$

quantity, which is called (i) the α-divergence [5], or (ii) for $\alpha=\frac{1}{2}$ the Bhattacharyya coefficient [12] (or the Bhattacharyya kernel, or the Hellinger affinity; see (56), (57) and (100)):

$$BC=\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}\in[0,1]. \tag{72}$$

(71) can also be further generalized to

$$D_{\mathrm{temp2}}(a,b)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^a\left[f_2(\mathbf{u})\right]^b f_1(\mathbf{u})\mathrm{d}\mathbf{u},\quad(a,b\in\mathbb{R}). \tag{73}$$

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 6 (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
```

Beyond the Rényi divergence $D_{\mathrm{R},\alpha}$ [122, 120, 124] ('Renyi_kNN_k'), the k-nearest neighbor technique can also be used to estimate the L_2- (D_{L}) [122, 120, 124] ('L2_kNN_k'), the Tsallis ($D_{\mathrm{T},\alpha}$) divergence [122, 120] ('Tsallis_kNN_k'), and of course, specifically to the Kullback-Leibler divergence (D) [81, 114, 190] ('KL_kNN_k', 'KL_kNN_kiTi'). A similar approach can be applied to the estimation of the (73) quantity [125], specifically to the Hellinger- and the Bhattacharyya distance ('Hellinger_kNN_k', 'Bhattacharyya_kNN_k'). For the MMD measure [49], (i) an U-statistic based ('MMD_Ustat'), (ii) a V-statistic based ('MMD_Vstat'), (iii) their incomplete Cholesky decomposition based accelerations ('MMD_Ustat_iChol', 'MMD_Vstat_iChol'), and (iv) a linearly scaling, online method ('MMD_online') have been implemented in ITE. The Cauchy-Schwartz and the Euclidean distance based divergences (D_{CS}, D_{ED}) can be estimated using KDE based plug-in methods, applying incomplete Cholesky decomposition ('CS_KDE_iChol', ''ED_KDE_iChol'). The energy distance (D_{EnDist}) can be estimated based on pairwise distances of sample points ('EnergyDist'). The Bregman distance and its symmetric variant can be estimated via k-nearest neighbors ('Bregman_kNN_k', 'symBregman_kNN_k'). The χ^2 distance D_{χ^2} can be estimated via (73), for which there are (i) k-nearest neighbor methods ('ChiSquare_kNN_k'), (ii) analytical formulas [104] in the exponential family ('ChiSquare_expF'). A power spectral density and Szegő's theorem based estimator for the Kullback-Leibler divergence is also available in ITE ('KL_PSD_SzegoT'). The Sharma-Mittal divergence can be estimated in ITE (i) using maximum likelihood estimation and analytical formula in the given exponential family [103] ('Sharma_expF'), or (ii) k-nearest neighbor methods [122, 120] ('Sharma_kNN_k'). Kullback-Leibler divergence estimation in the exponential family can be carried out making use of Bregman distances ('KL_expF'). Von Mises expansion has been adapted to estimate Kullback-Leibler-, Rényi, Tsallis-, Pearson χ^2 divergence and Hellinger distance [72] ('KL_vME', 'Renyi_vME', 'Tsallis_vME', 'ChiSquare_vME', 'Hellinger_vME').

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 6 to the cost_name given in the last column of the table.

3.1.4 Association Measure Estimators

There exist many exciting association quantities measuring certain dependency relations of random variables, for a recent excellent review on the topic, see [136]. 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 $\boldsymbol{\rho}$** (also called the Spearman's rank correlation coefficient, or the grade correlation coefficient) [150]. For $d = 2$, it is defined as

$$A_\rho\left(y^1, y^2\right) = corr\left(F_1\left(y^1\right), F_2\left(y^2\right)\right), \tag{74}$$

where 'corr' stands for correlation and F_i denotes the (cumulative) distribution function (cdf) of y^i. Spearman's ρ 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 A_ρ. For a formal definition of measures of concordance, see Def. 2 (Section D).

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

$$M(\mathbf{u}) = \min_{i=1,\ldots,d} u_i. \tag{75}$$

The name originates from the fact that for any C copula

$$W(\mathbf{u}) := \max(u_1 + \ldots + u_d - d + 1, 0) \leq C(\mathbf{u}) \leq M(\mathbf{u}), \quad (\forall \mathbf{u} \in [0,1]^d) \tag{76}$$

Here, W is called the Fréchet-Hoeffding lower bound.[15]

It is known that A_ρ can be interpreted as the normalized average difference of the copula of $\mathbf{y}$ (C) and the independence copula (Π) [see Eq. (22)]:

$$A_\rho\left(y^1, y^2\right) = A_\rho(C) = \frac{\int_{[0,1]^2} u_1 u_2 \mathrm{d}C(\mathbf{u}) - \left(\frac{1}{2}\right)^2}{\frac{1}{12}} = 12 \int_{[0,1]^2} C(\mathbf{u})\mathrm{d}\mathbf{u} - 3 = \frac{\int_{[0,1]^2} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^2} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^2} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^2} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}, \tag{77}$$

[15] W is a copula only in two dimensions ($d = 2$).

Estimated quantity	Principle	d	`cost_name`
L_2 divergence (D_{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_{MMD})	U-statistics, unbiased	$d \geq 1$	`'MMD_Ustat'`
maximum mean discrepancy (D_{MMD})	V-statistics, biased	$d \geq 1$	`'MMD_Vstat'`
maximum mean discrepancy (D_{MMD})	online	$d \geq 1$	`'MMD_online'`
maximum mean discrepancy (D_{MMD})	U-statistics, incomplete Cholesky decomposition	$d \geq 1$	`'MMD_Ustat_iChol'`
maximum mean discrepancy (D_{MMD})	V-statistics, incomplete Cholesky decomposition	$d \geq 1$	`'MMD_Vstat_iChol'`
Hellinger distance (D_{H})	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'Hellinger_kNN_k'`
Bhattacharyya distance (D_{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_{CS})	KDE, incomplete Cholesky decomposition	$d \geq 1$	`'CS_KDE_iChol'`
Euclidean distance based divergence (D_{ED})	KDE, incomplete Cholesky decomposition	$d \geq 1$	`'ED_KDE_iChol'`
energy distance (D_{EnDist})	pairwise distances	$d \geq 1$	`'EnergyDist'`
Bregman distance ($D_{\mathrm{NB},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'Bregman_kNN_k'`
symmetric Bregman distance ($D_{\mathrm{SB},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'symBregman_kNN_k'`
Pearson χ^2 divergence (D_{χ^2})	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'ChiSquare_kNN_k'`
Kullback-Leibler divergence (D)	power spectral density, Szegő's theorem	$d = 1$	`'KL_PSD_SzegoT'`
Sharma-Mittal divergence ($D_{\mathrm{SM},\alpha,\beta}$)	MLE + analytical value in the exponential family	$d \geq 1$	`'Sharma_expF'`
Sharma-Mittal divergence ($D_{\mathrm{SM},\alpha,\beta}$)	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'SharmaM_kNN_k'`
Kullback-Leibler divergence (D)	MLE + analytical value in the exponential family	$d \geq 1$	`'KL_expF'`
Pearson χ^2 divergence (D_{χ^2})	MLE + analytical value in the exponential family	$d \geq 1$	`'ChiSquare_expF'`
Kullback-Leibler divergence (D)	von Mises expansion	$d \geq 1$	`'KL_vME'`
Rényi divergence ($D_{\mathrm{R},\alpha}$)	von Mises expansion	$d \geq 1$	`'Renyi_vME'`
Tsallis divergence ($D_{\mathrm{T},\alpha}$)	von Mises expansion	$d \geq 1$	`'Tsallis_vME'`
Pearson χ^2 divergence (D_{χ^2})	von Mises expansion	$d \geq 1$	`'ChiSquare_vME'`
Hellinger distance (D_{H})	von Mises expansion	$d \geq 1$	`'Hellinger_vME'`

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

where the

$$\int_{[0,1]^2} M(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{1}{3}, \qquad \int_{[0,1]^2} \Pi(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{1}{4} \tag{78}$$

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

1. **Spearman's ρ, multivariate-1**: One can extend [194, 67, 96, 135] the Spearman's ρ to the multivariate case using (77) as

$$A_{\rho_1}\left(y^1,\ldots,y^d\right) = A_{\rho_1}(C) = \frac{\int_{[0,1]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}} = h_\rho(d)\left[2^d \int_{[0,1]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - 1\right], \tag{79}$$

   where

$$h_\rho(d) = \frac{d+1}{2^d - (d+1)}. \tag{80}$$

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

   Note:

   - A_{ρ_1} satisfies all the axioms of multivariate measure of concordance (see Def. 3 in Section D) except for Duality [177].
   - A_{ρ_1} can also be derived from average *lower orthant dependence* ideas [96].

2. **Spearman's ρ, multivariate-2**: An other multivariate extension [67, 96, 135] of Spearman's ρ is using (77)

$$A_{\rho_2}\left(y^1,\ldots,y^d\right) = A_{\rho_2}(C) = \frac{\int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}C(\mathbf{u}) - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}} = h_\rho(d)\left[2^d \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}C(\mathbf{u}) - 1\right]. \tag{81}$$

   The association measure is called '`Spearman2`' in ITE.

   Note:

   - A_{ρ_2} satisfies all the axioms of multivariate measure of concordance (see Def. 3 in Section D) except for Duality [177].
   - A_{ρ_2} can also be derived using an average *upper orthant dependence* approach [96].

3. **Spearman's ρ, multivariate-3**: [97, 98] further considers the average of A_{ρ_1} and A_{ρ_2}, i.e.

$$A_{\rho_3}\left(y^1,\ldots,y^d\right) = A_{\rho_3}(C) = \frac{A_{\rho_1}\left(y^1,\ldots,y^d\right) + A_{\rho_2}\left(y^1,\ldots,y^d\right)}{2}. \tag{82}$$

   The name of this association measure is '`Spearman3`' in ITE.[16]

   Note:

   - For the special case of $d = 2$, the defined extensions of Spearman's ρ coincide:

$$A_\rho = A_{\rho_1} = A_{\rho_2} = A_{\rho_3}. \tag{83}$$

   - A_{ρ_3} is a multivariate measure of concordance (see Def. 3 in Section D).

4. **Spearman's ρ, multivariate-4**: The average pairwise Spearman's ρ is defined [73, 135] as

$$A_{\rho_4}\left(y^1,\ldots,y^d\right) = A_{\rho_4}(C) = h(2)\left[2^2\binom{d}{2}^{-1}\sum_{k,l=1;k<l}^{d}\int_{[0,1]^2} C_{kl}(u,v)\mathrm{d}u\mathrm{d}v - 1\right] = \binom{d}{2}^{-1}\sum_{k,l=1;k<l}^{d} A_\rho\left(y^k,y^l\right), \tag{84}$$

   where C_{kl} denotes the bivariate marginal copula of C corresponding to the k^{th} and l^{th} margin. The name of the association measure is '`Spearman4`' in ITE. A_{ρ_4} is a multivariate measure of concordance (see Def. 3 in Section D).

[16] Although (82) would make it possibile to implement A_{ρ_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.

5. **Correntropy**, **centered correntropy**, **correntropy coefficient** [128]: These association measures are defined as

$$A_{\mathrm{CorrEntr}}\left(y^1,y^2\right)=\mathbb{E}_{y^1,y^2}\left[k\left(y^1,y^2\right)\right]=\int_{\mathbb{R}^2}k(u,v)\mathrm{d}F_{y^1,y^2}(u,v), \tag{85}$$

$$A_{\mathrm{CCorrEntr}}\left(y^1,y^2\right)=\mathbb{E}_{y^1,y^2}\left[k\left(y^1,y^2\right)\right]-\mathbb{E}_{y^1}\mathbb{E}_{y^2}\left[k\left(y^1,y^2\right)\right]=\int_{\mathbb{R}^2}k(u,v)\left[\mathrm{d}F_{y^1,y^2}(u,v)-\mathrm{d}F_{y^1}\mathrm{d}F_{y^2}(u,v)\right], \tag{86}$$

$$A_{\mathrm{CorrEntrCoeff}}\left(y^1,y^2\right)=\frac{A_{\mathrm{CCorrEntr}}\left(y^1,y^2\right)}{\sqrt{A_{\mathrm{CCorrEntr}}\left(y^1,y^1\right)}\sqrt{A_{\mathrm{CCorrEntr}}\left(y^2,y^2\right)}}\in[-1,1], \tag{87}$$

where F_{y^1,y^2} (F_{y^i}) stands for the distribution function of $\mathbf{y}=\left[y^1;y^2\right]$ (y^i) and k is a kernel. Specially, for $k(u,v)=uv$ the centered correntropy reduces to the covariance, and the correntropy coefficient to the traditional correlation coefficient. The name of the estimators in ITE are '`CorrEntr_KDE_direct`', '`CCorrEntr_KDE_iChol`', '`CCorrEntr_KDE_Lapl`', '`CorrEntrCoeff_KDE_direct`', and '`CorrEntrCoeff_KDE_iChol`'.

6. **Multivariate extension of Blomqvist's β (medial correlation coefficient)**: Let $\mathbf{y}\in\mathbb{R}^2$, and let $\tilde{y}^i$ be the median of y^i. Blomqvist's β is defined [93, 14] as

$$A_\beta\left(y^1,y^2\right)=\mathbb{P}\left(\left(y^1-\tilde{y}^1\right)\left(y^2-\tilde{y}^2\right)>0\right)-\mathbb{P}\left(\left(y^1-\tilde{y}^1\right)\left(y^2-\tilde{y}^2\right)<0\right). \tag{88}$$

It can be expressed in terms of the C, the copula of $\mathbf{y}$:

$$A_\beta\left(y^1,y^2\right)=A_\beta(C)=4C\left(\frac{1}{2},\frac{1}{2}\right)-1=\frac{C\left(\frac{1}{2},\frac{1}{2}\right)-\Pi\left(\frac{1}{2},\frac{1}{2}\right)+\bar{C}\left(\frac{1}{2},\frac{1}{2}\right)-\bar{\Pi}\left(\frac{1}{2},\frac{1}{2}\right)}{M\left(\frac{1}{2},\frac{1}{2}\right)-\Pi\left(\frac{1}{2},\frac{1}{2}\right)+\bar{M}\left(\frac{1}{2},\frac{1}{2}\right)-\bar{\Pi}\left(\frac{1}{2},\frac{1}{2}\right)}. \tag{89}$$

where $\bar{C}(\mathbf{u})$ denotes the *survival function*[17]:

$$\bar{C}(\mathbf{u}):=\mathbb{P}(\mathbf{U}>\mathbf{u}),\quad(\mathbf{u}=[u_1;\ldots;u_d]\in[0,1]^d). \tag{90}$$

A_β [Eq. (88)] is a measure of concordance (see Def. 2 in Section D). A natural multivariate ($\mathbf{y}\in\mathbb{R}^d$, $d_m=1$) generalization [182, 136] of Blomqvist's β motivated by (89) is

$$A_\beta\left(y^1,\ldots,y^d\right)=A_\beta(C)=\frac{C\left(\mathbf{1/2}\right)-\Pi\left(\mathbf{1/2}\right)+\bar{C}\left(\mathbf{1/2}\right)-\bar{\Pi}\left(\mathbf{1/2}\right)}{M\left(\mathbf{1/2}\right)-\Pi\left(\mathbf{1/2}\right)+\bar{M}\left(\mathbf{1/2}\right)-\bar{\Pi}\left(\mathbf{1/2}\right)}=h_\beta(d)\left[C(\mathbf{1/2})+\bar{C}(\mathbf{1/2})-2^{1-d}\right], \tag{91}$$

where $\mathbf{1/2}=\left[\frac{1}{2};\ldots;\frac{1}{2}\right]\in\mathbb{R}^d$ and

$$h_\beta(d)=\frac{2^{d-1}}{2^{d-1}-1}. \tag{92}$$

The objective [Eq. (91)] is called '`Blomqvist`' in ITE.[18] A_β [Eq. (91)] satisfies all the axioms of multivariate measure of concordance (see Def. 3 in Section D) except for Duality [177].

7. **Multivariate conditional version of Spearman's ρ (lower/upper tail)**: Let g be a non-negative function, for which the following integral exists [134]:

$$A_{\rho_g}\left(y^1,\ldots,y^d\right)=A_{\rho_g}(C)=\frac{\int_{[0,1]^d}C(\mathbf{u})g(\mathbf{u})\mathrm{d}\mathbf{u}-\int_{[0,1]^d}\Pi(\mathbf{u})g(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^d}M(\mathbf{u})g(\mathbf{u})\mathrm{d}\mathbf{u}-\int_{[0,1]^d}\Pi(\mathbf{u})g(\mathbf{u})\mathrm{d}\mathbf{u}}. \tag{93}$$

Here, g is a weighting function, emphasizing specific parts of the copula.

   (a) **Lower tail**: Specially, let $g(\mathbf{u})=\mathbb{I}_{[0,p]^d}(\mathbf{u})$ $(0<p\leq 1)$, where $\mathbb{I}$ stands for the indicator function. This g choice refers to the weighting of the lower part of the copula, i.e., we measure the amount of depedence in the lower tail of the multivariate distributions.

[17] $\bar{C}$ is not in general a copula.

[18] Despite Eq. (91), '`Blomqvist`' is implemented as a base association measure estimator to avoid the computation of the same rank statistics multiple times.

Estimated quantity	Principle	d_m	M	cost_name
Spearman's ρ: multivariate1 (A_{ρ_1})	empirical copula, explicit formula	$d_m = 1$	$M \geq 2$	'Spearman1'
Spearman's ρ: multivariate2 (A_{ρ_2})	empirical copula, explicit formula	$d_m = 1$	$M \geq 2$	'Spearman2'
Spearman's ρ: multivariate3 (A_{ρ_3})	ρ_3 is the average of ρ_1 and ρ_2	$d_m = 1$	$M \geq 2$	'Spearman3'
Spearman's ρ: multivariate4 (A_{ρ_4})	average pairwise Spearman's ρ	$d_m = 1$	$M \geq 2$	'Spearman4'
correntropy (A_{CorrEntr})	KDE, direct	$d_m = 1$	$M = 2$	'CorrEntr_KDE_direct'
centered correntropy ($A_{\text{CCorrEntr}}$)	KDE, incomplete Cholesky decomp.	$d_m = 1$	$M = 2$	'CCorrEntr_KDE_iChol'
centered correntropy ($A_{\text{CCorrEntr}}$)	KDE, Laplacian kernel, sorting	$d_m = 1$	$M = 2$	'CCorrEntr_KDE_Lapl'
correntropy coefficient ($A_{\text{CorrEntrCoeff}}$)	KDE, direct	$d_m = 1$	$M = 2$	'CorrEntrCoeff_KDE_direct'
correntropy coefficient ($A_{\text{CorrEntrCoeff}}$)	KDE, incomplete Cholesky decomp.	$d_m = 1$	$M = 2$	'CorrEntrCoeff_KDE_iChol'
Blomqvist's β (A_β)	empirical copula, explicit formula	$d_m = 1$	$M \geq 2$	'Blomqvist'
conditional Spearman's ρ, lower tail ($A_{\rho_{\text{lt}}}$)	empirical copula, explicit formula	$d_m = 1$	$M \geq 2$	'Spearman_lt'
conditional Spearman's ρ, upper tail ($A_{\rho_{\text{ut}}}$)	empirical copula, explicit formula	$d_m = 1$	$M \geq 2$	'Spearman_ut'

Table 5: Association measure 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; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$).

The resulting conditional version of Spearman's ρ is

$$A_{\rho_{\text{lt}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\text{lt}}}(C) = \frac{\int_{[0,p]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,p]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,p]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,p]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}} = \frac{\int_{[0,p]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \left(\frac{p^2}{2}\right)^d}{\frac{p^{d+1}}{d+1} - \left(\frac{p^2}{2}\right)^d}. \tag{94}$$

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

Note:

- Specially, for $p = 1$ the association $A_{\rho_{\text{lt}}}$ reduces to A_{ρ_1} [Eq. (79)].
- One can show that $A_{\rho_{lt}}$ preserves the concordance ordering [see Eq. (258)], i.e., $C_1 \prec C_2 \Rightarrow A_{\rho_{lt}}(C_1) \leq A_{\rho_{lt}}(C_2)$, for $\forall p \in (0, 1]$. Specially, from $C \prec M$ [see Eq. (76)] one obtains that $A_{\rho_{lt}} \leq 1$.

(b) **Upper tail:** In this case, in Eq. (93) our choice is $g(\mathbf{u}) = \mathbb{I}_{[1-p,1]^d}(\mathbf{u})$ $(0 < p \leq 1)$, i.e., the upper tail of the copula is weighted:

$$A_{\rho_{\text{ut}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\text{ut}}}(C) = \frac{\int_{[1-p,1]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[1-p,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[1-p,1]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[1-p,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}. \tag{95}$$

The name of the objective is 'Spearman_ut' in ITE.

The calling syntax of the association measure estimators is unified and very simple; as an example the A_{ρ_1} measure is estimated:

Example 7 (Association measure 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 measure 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 measure estimators.

3.1.5 Cross Quantity 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.

Estimated quantity	Principle	d	`cost_name`
cross-entropy (C_{CE})	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'CE_kNN_k'`
cross-entropy (C_{CE})	MLE + analytical value in the exponential family	$d \geq 1$	`'CE_expF'`

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

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:

$$C_{\mathrm{CE}}(f_1, f_2) = -\int_{\mathbb{R}^d} f_1(\mathbf{u}) \log\left[f_2(\mathbf{u})\right] \mathrm{d}\mathbf{u}. \tag{96}$$

One can estimate C_{CE} via the k-nearest neighbor ($S = \{k\}$) technique [81] (`'CE_kNN_k'`) or by MLE combined with analytical formula in the chosen exponential family [102] (`'CE_expF'`).

The calling syntax of the cross quantity estimators is unified, an example is given below:

Example 8 (Cross quantity 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 = 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 quantity estimators of ITE are summarized in Table 6.

3.1.6 Estimators of Kernels on Distributions

Kernels on distributions quantify the 'similarity' of ν_1 and ν_2, two distributions (probability measures) on a given $\mathcal{X}$ space ($\nu_1, \nu_2 \in \mathcal{M}_+^1(\mathcal{X})$). By definition the $K : \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{X}) \to \mathbb{R}$ kernel is symmetric and positive definite, i.e.,

1. $K(\nu_1, \nu_2) = K(\nu_2, \nu_1)$ $(\forall \nu_1, \nu_2 \in \mathcal{M}_+^1(\mathcal{X}))$, and
2. $\sum_{i,j=1}^n c_i c_j K(\nu_i, \nu_j) \geq 0$, for all n positive number, $\{c_i\}_{i=1}^n \in \mathbb{R}^n$ and $\{\nu_i\}_{i=1}^n \in \left[\mathcal{M}_+^1(\mathcal{X})\right]^n$.

An alternative, equivalent view of kernels is that they compute the inner product of their arguments embedded into a suitable Hilbert space ($\mathcal{H}$). In other words, there exist a $\varphi : \mathcal{M}_+^1(\mathcal{X}) \to \mathcal{H}$ mapping, where $\mathcal{H}$ is a Hilbert space such that

$$K(\nu_1, \nu_2) = \langle \varphi(\nu_1), \varphi(\nu_2) \rangle_{\mathcal{H}}, \quad (\forall \nu_1, \nu_2 \in \mathcal{M}_+^1(\mathcal{X})). \tag{97}$$

In the simplest case $\mathcal{X} = \mathbb{R}^d$ and the ν_1, ν_2 distributions are identified with their densities $f_1 : \mathbb{R}^d \mapsto \mathbb{R}$ and $f_2 : \mathbb{R}^d \mapsto \mathbb{R}$. Our goal is to estimate the value of the kernel $[K(f_1, f_2)]$ given 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. It is also worth noting that many widely used divergences (see Section 3.1.3 and 3.1.3) can be induced by kernels, see [56].

ITE can estimate the following kernels on distributions:

1. **Expected kernel** [54, 43, 50]: The expected kernel (also called the summation kernel, the mean map kernel, the set kernel, the multi-instance kernel, the ensemble kernel; a special convolution kernel) is the inner product of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, the mean embedding [11] of f_1 and f_2 [see (52)]

$$K_{\exp}(f_1, f_2) = \langle \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \rangle_{\mathcal{F}} = \mathbb{E}_{\mathbf{y}^1, \mathbf{y}^2}\left[k\left(\mathbf{y}^1, \mathbf{y}^2\right)\right], \tag{98}$$

   i.e., it generates MMD

$$\left[D_{\mathrm{MMD}}(f_1, f_2)\right]^2 = K_{\exp}(f_1, f_1) - 2K_{\exp}(f_1, f_2) + K_{\exp}(f_1, f_2). \tag{99}$$

   The estimator of the expected kernel is called `'expected'` in ITE.

Estimated quantity	Principle	d	`cost_name`
expected kernel ($K_{\exp}$)	mean of pairwise kernel values	$d \geq 1$	`'expected'`
Bhattacharyya kernel (K_{B})	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'Bhattacharyya_kNN_k'`
probability product kernel ($K_{\mathrm{PP},\rho}$)	k-nearest neighbors ($S = \{k\}$)	$d \geq 1$	`'PP_kNN_k'`

Table 7: Estimators of kernels on distributions (base). Third column: dimension (d) constraint.

2. **Bhattacharyya kernel**: The Bhattacharyya kernel [12, 65] (also known as the Bhattacharyya coefficient, or the Hellinger affinity; see (72)) is defined as

$$K_{\mathrm{B}}(f_1, f_2) = \int_{\mathbb{R}^d} \sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}. \tag{100}$$

   It

   - is intimately related to, induces the Hellinger distance [see Eq. (56)]:

$$[D_{\mathrm{H}}(f_1, f_2)]^2 = \frac{1}{2}\left[K_{\mathrm{B}}(f_1, f_1) - 2K_{\mathrm{B}}(f_1, f_2) + K_{\mathrm{B}}(f_2, f_2)\right] = \frac{1}{2}[2 - 2K_{\mathrm{B}}(f_1, f_2)] = 1 - K_{\mathrm{B}}(f_1, f_2)]. \tag{101}$$

   - is sufficient to estimate (73), for which there exist k-nearest neighbor methods [125]. The associated estimator is called '`Bhattacharyya_kNN_k`' in ITE.

3. **Probability product kernel:** The probability product kernel [65] is the inner product of the ρ^{th} power of the densities

$$K_{\mathrm{PP},\rho}(f_1, f_2) = \int_{\mathbb{R}^d} [f_1(\mathbf{u})]^\rho [f_2(\mathbf{u})]^\rho \,\mathrm{d}\mathbf{u}, \quad (\rho > 0). \tag{102}$$

   Notes:

   - Specially, for $\rho = \frac{1}{2}$ we get back the Bhattacharyya kernel [see (100)].
   - It is sufficient to estimate the (73) quantity, for which k-nearest neighbor techniques are available [125]. The corresponding estimator is called '`PP_kNN_k`' in ITE.

The calling syntax of kernels on distributions is unified. In the following example the estimation of the expected kernel is illustrated:

Example 9 (Kernel estimation on distributions (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 = Kexpected_initialization(mult); %initialize the kernel ('K') estimator on
                                       %distributions ('expected')
>K = Kexpected_estimation(Y1,Y2,co);  %perform kernel estimation on distributions
```

The available base kernel estimators on distributions are enlisted in Table 7; for the estimation of other kernels it is enough to change '`expected`' to the `cost_name` given in the last column of the table.

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 measure and cross quantity estimators, respectively.

3.2.1 Entropy Estimators

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

1. **Ensemble**: 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 [77]. 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

$$H_{\text{ensemble}}(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \hat{H}\left(\{\mathbf{y}_t\}_{t \in I_n}\right). \tag{103}$$

   As a prototype example for meta entropy estimation the implementation of the ensemble method [Eq. (103)] is provided below (see Example 10 and Example 11). 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 (103)] to *any* other entropy estimation technique (base/meta, see Table 2 and Table 8).

   Example 10 (Entropy estimation (meta: initialization))

```
function [co] = Hensemble_initialization(mult)
co.name = 'ensemble';             %name of the estimator: 'ensemble'
co.mult = 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 (103):

   Example 11 (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 3, Example 4):

   Example 12 (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. **Random projected ensemble**: Since (i) entropy can be estimated consistently using pairwise distances of sample points[19], and (ii) random projection (RP) techniques realize approximate isometric embeddings [68, 38, 64, 1, 82, 6, 90], one can construct efficient estimation methods by the integration of the ensemble and the RP technique.

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

Formally, the definition of the estimation is identical to that of the ensemble approach [Eq. (103)], except for random projections $\mathbf{R}_n \in \mathbb{R}^{d_{RP} \times d}$ $(n = 1, \ldots, N)$. The final estimation is

$$H_{\text{RPensemble}}(\mathbf{y}) = \frac{1}{N} \sum_{n=1}^{N} \hat{H} \left(\{\mathbf{R}_n \mathbf{y}_t\}_{t \in I_n} \right). \tag{104}$$

The approach shows exciting potentials with serious computational speed-ups in independent subspace analysis [163] and image registration [164]. The technique has been implemented in the ITE toolbox under the name '`RPensemble`'.

3. **Complex**: Information theoretical quantities can be defined over the complex domain via the Hilbert transformation [36]

$$\varphi_v : \mathbb{C}^d \ni \mathbf{v} \mapsto \mathbf{v} \otimes \begin{bmatrix} \Re(\cdot) \\ \Im(\cdot) \end{bmatrix} \in \mathbb{R}^{2d}, \tag{105}$$

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

$$H_{\mathbb{C}}(\mathbf{y}) := H(\varphi_v(\mathbf{y})). \tag{106}$$

Relation (106) can be transformed to a meta entropy estimator, the method is available under the name '`complex`'.

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

$$H_{\text{T},\alpha}(\mathbf{y}) = \frac{e^{(1-\alpha) H_{\text{R},\alpha}(\mathbf{y})} - 1}{1 - \alpha}. \tag{107}$$

The formula is realized in ITE by the '`Tsallis_HRenyi`' meta entropy estimator. Making use of this approach, for example, the Rényi entropy estimators of Table 2 can be instantly applied for Tsallis entropy estimation.

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

$$\mathbf{y}_G \sim f_G = N(\mathbb{E}(\mathbf{y}), cov(\mathbf{y})). \tag{108}$$

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

$$H(\mathbf{y}_G) = \frac{1}{2} \log \left[(2\pi e)^d \det(cov(\mathbf{y})) \right], \tag{109}$$

moreover, $H(\mathbf{y})$ equals to $H(\mathbf{y}_G)$ minus the Kullback-Leibler divergence [see Eq. (44)] of $\mathbf{y} \sim f$ and f_G [191]:

$$H(\mathbf{y}) = H(\mathbf{y}_G) - D(f, f_G). \tag{110}$$

The associated meta entropy estimator is called '`Shannon_DKL_N`'.

6. **Divergence from the uniform distribution**: If $\mathbf{y} \in [0,1]^d$ $(\sim f)$, then the entropy of $\mathbf{y}$ equals to minus the Kullback-Leibler divergence [see Eq. (44)] of f and f_U, the uniform distribution on $[0,1]^d$:

$$H(\mathbf{y}) = -D(f, f_U). \tag{111}$$

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 [25], one obtains that

$$H(\mathbf{y}) = -D(f', f_U) + \log \left[\prod_{i=1}^{d} (b_i - a_i) \right]. \tag{112}$$

This meta entropy estimation technique is called '`Shannon_DKL_U`' in ITE.

The meta entropy estimator methods in ITE are summarized in Table 8. The calling syntax of the estimators is identical to Example 12, one only has to change the name '`ensemble`' to the `cost_name` of the target estimators, see the last column of the table.

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 8: Entropy estimators (meta). Third column: dimension (d) constraint.

3.2.2 Mutual Information Estimators

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

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

   Example 13 (Mutual information estimator (meta: initialization))

```
function [co] = IShannon_HShannon_initialization(mult)
co.name = 'Shannon_HShannon';      %name of the estimator: 'Shannon_HShannon'
co.mult = 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 14 (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 5):

   Example 15 (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. **Complex**: The mutual information of complex random variables ($\mathbf{y} \in \mathbb{C}^{d_m}$) can be defined via the Hilbert transformation [Eq. (105)]:

$$I_{\mathbb{C}}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = I\left(\varphi_v\left(\mathbf{y}^1\right),\ldots,\varphi_v\left(\mathbf{y}^M\right)\right). \tag{113}$$

   The relation is realized in ITE by the 'complex' meta estimator.

3. **Shannon-, L_2-, Tsallis- and Rényi mutual information**: 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 ($\prod_{m=1}^{M} f_m$)[20]:

$$I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D\left(f, \prod_{m=1}^{M} f_m\right), \qquad I_{\mathrm{L}}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{L}}\left(f, \prod_{m=1}^{M} f_m\right), \tag{114}$$

$$I_{\mathrm{T},\alpha}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{T},\alpha}\left(f, \prod_{m=1}^{M} f_m\right), \qquad I_{\mathrm{R},\alpha}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{R},\alpha}\left(f, \prod_{m=1}^{M} f_m\right). \tag{115}$$

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

$$I_{\mathrm{R},\alpha} \xrightarrow{\alpha\to 1} I, \qquad I_{\mathrm{T},\alpha} \xrightarrow{\alpha\to 1} I. \tag{116}$$

The associated Shannon-, Rényi-, L_2- and Tsallis meta mutual information estimators are available in ITE using the names '`Shannon_DKL`', '`Renyi_DRenyi`', '`L2_DL2`' and '`Tsallis_DTsallis`'. The Rényi mutual information of one-dimensional variables ($d_m = 1$, $\forall m$) can also be expressed [109, 110] as minus the Rényi entropy of the joint copula, i.e.,

$$I_{\mathrm{R},\alpha}\left(y^1,\ldots,y^M\right) = -H_{\mathrm{R},\alpha}(\mathbf{Z}), \tag{117}$$

where

$$\mathbf{Z} = \left[F_1\left(y^1\right);\ldots;F_M\left(y^M\right)\right] \in \mathbb{R}^M \tag{118}$$

is the joint copula, F_m is the cumulative density function of y^m. The meta estimator is called `Renyi_HRenyi` in ITE.

4. **Copula based kernel dependency**: [115] 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. (52)] of the joint copula and the M-dimensional uniform distribution on $[0,1]^M$:

$$I_{\mathrm{c}}\left(y^1,\ldots,y^M\right) = D_{\mathrm{MMD}}(\mathbb{P}_{\mathbf{Z}},\mathbb{P}_{\mathbf{U}}), \tag{119}$$

where we used the same notation as in Eq. (118) and $\mathbb{P}$ denotes the distribution. The associated meta estimator has the name '`MMD_DMMD`' in ITE.

5. **Distance covariance**: An alternative form of the distance covariance [Eq. (34) and $\alpha = 1$] in terms of pairwise distances is

$$\begin{aligned} I_{\mathrm{dCov}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = {} & \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{1'},\mathbf{y}^{2'}}\left[\left\|\mathbf{y}^1-\mathbf{y}^{1'}\right\|_2\left\|\mathbf{y}^2-\mathbf{y}^{2'}\right\|_2\right] + \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^{1'}}\left[\left\|\mathbf{y}^1-\mathbf{y}^{1'}\right\|_2\right]\mathbb{E}_{\mathbf{y}^2,\mathbf{y}^{2'}}\left[\left\|\mathbf{y}^2-\mathbf{y}^{2'}\right\|_2\right] \\ & - 2\mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\left[\mathbb{E}_{\mathbf{y}^{1'}}\left\|\mathbf{y}^1-\mathbf{y}^{1'}\right\|_2\mathbb{E}_{\mathbf{y}^{2'}}\left\|\mathbf{y}^2-\mathbf{y}^{2'}\right\|_2\right], \end{aligned} \tag{120}$$

where $(\mathbf{y}^1,\mathbf{y}^2)$ and $(\mathbf{y}^{1'},\mathbf{y}^{2'})$ are i.i.d. variables. The concept of distance covariance and the formula above can also be extended to semimetric spaces $[(\mathcal{Y}_1,\rho_1),(\mathcal{Y}_2,\rho_2)]$ of negative type [86, 140] (see Def. 6, Section D):

$$\begin{aligned} I_{\mathrm{dCov}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = {} & \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\mathbb{E}_{\mathbf{y}^{1'},\mathbf{y}^{2'}}\left[\rho_1\left(\mathbf{y}^1,\mathbf{y}^{1'}\right)\rho_2\left(\mathbf{y}^2,\mathbf{y}^{2'}\right)\right] + \mathbb{E}_{\mathbf{y}^1,\mathbf{y}^{1'}}\left[\rho_1\left(\mathbf{y}^1,\mathbf{y}^{1'}\right)\right]\mathbb{E}_{\mathbf{y}^2,\mathbf{y}^{2'}}\left[\rho_2\left(\mathbf{y}^2,\mathbf{y}^{2'}\right)\right] \\ & - 2\mathbb{E}_{\mathbf{y}^1,\mathbf{y}^2}\left(\mathbb{E}_{\mathbf{y}^{1'}}\left[\rho_1\left(\mathbf{y}^1,\mathbf{y}^{1'}\right)\right]\mathbb{E}_{\mathbf{y}^{2'}}\left[\left(\mathbf{y}^2,\mathbf{y}^{2'}\right)\right]\right). \end{aligned} \tag{121}$$

The resulting measure can be proved to be expressible in terms of HSIC [Eq. (16)]:

$$\left[I_{\mathrm{dCov}}\left(\mathbf{y}^1,\mathbf{y}^2\right)\right]^2 = 4[D_{\mathrm{MMD}}(f,f_1f_2)]^2 = 4\left[I_{\mathrm{HSIC}}\left(\mathbf{y}^1,\mathbf{y}^2\right)\right]^2, \tag{122}$$

where the kernel k (used in HSIC) is

$$k((\mathbf{u}_1,\mathbf{v}_1),(\mathbf{u}_2,\mathbf{v}_2)) = k_1(\mathbf{u}_1,\mathbf{u}_2)k_2(\mathbf{v}_1,\mathbf{v}_2) \tag{123}$$

with k_i kernels generating [see Eq. (134)] ρ_i-s ($i = 1,2$). The meta estimator is called '`dCov_IHSIC`' in ITE.

[20] For the definitions of f and f_ms, see Eq. (10). The divergence definitions can be found in Eqs. (44), (45), (46) and (49).

6. **Approximate correntropy independence measure** [128]: This measure is defined as

$$I_{\text{ACorrEntr}}\left(y^1,y^2\right)=\max\left[\left|A_{\text{CCorrEntr}}\left(y^1,y^2\right)\right|,\left|A_{\text{CCorrEntr}}\left(-y^1,y^2\right)\right|\right]. \tag{124}$$

The meta mutual information estimator is available in ITE under the name '`ApprCorrEntr`'.

Note: the correntropy independence measure

$$\sup_{a,b\in\mathbb{R}}\left|U_{a,b}\left(y^1,y^2\right)\right|, \tag{125}$$

where

$$U_{a,b}\left(y^1,y^2\right)=A_{\text{CCorrEntr}}\left(ay^1+b,y^2\right)\quad(a\neq 0) \tag{126}$$

is a valid independence measure in the sense, that it [Eq. (125)] is zero if and only if y^1 and y^2 are independent. $I_{\text{ACorrEntr}}$ [Eq. (124)] is an approximation of this quantity in a bivariate mixture of Gaussian approach.

7. χ^2 **mutual information**: the χ^2 mutual information is defined as the Pearson χ^2 divergence [112] (see Eq. (68)) of the joint density and the product of the marginals

$$I_{\chi^2}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right)=D_{\chi^2}\left(f,\prod_{m=1}^{M}f_m\right). \tag{127}$$

The corresponding meta estimator is called '`ChiSquare_DChiSquare`' in ITE.

Notes: for two components ($M=2$), this measure is also referred to as the

- the Hilbert-Schmidt norm of the normalized cross-covariance operator [39, 40], i.e.,

$$I_{\chi^2}\left(\mathbf{y}^1,\mathbf{y}^2\right)=\left\|\mathbf{V}_{\mathbf{y}^2,\mathbf{y}^1}\right\|_{\text{HS}}^2, \tag{128}$$

  where $\mathbf{V}_{\mathbf{y}^2,\mathbf{y}^1}$ is defined via the decomposition of the cross-covariance operator [see Eq. (14)]

$$\mathbf{C}_{\mathbf{y}^2,\mathbf{y}^1}=\left(\mathbf{C}_{\mathbf{y}^2,\mathbf{y}^2}\right)^{\frac{1}{2}}\mathbf{V}_{\mathbf{y}^2,\mathbf{y}^1}\left(\mathbf{C}_{\mathbf{y}^1,\mathbf{y}^1}\right)^{\frac{1}{2}}. \tag{129}$$

- the squared-loss mutual information [155, 105], or
- the mean square contingency [131]

$$\left[I_{\text{MSC}}\left(\mathbf{y}^1,\mathbf{y}^2\right)\right]^2=I_{\chi^2}\left(\mathbf{y}^1,\mathbf{y}^2\right). \tag{130}$$

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 9. 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 10). Our prototype example is the estimation of the **symmetrised Kullback-Leibler divergence**, the so-called J-distance (or J divergence):

$$D_{\text{J}}(f_1,f_2)=D(f_1,f_2)+D(f_2,f_1). \tag{131}$$

The definition of meta divergence estimators follows the idea of meta entropy and mutual information estimators (see Example 10, 11, 13 and 14). Initialization and estimation of the meta J-distance estimator can be carried out as follows:

Example 16 (Divergence estimator (meta: initialization))

```
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
```

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

Table 9: 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; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$).

Example 17 (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 6).

Example 18 (Divergence estimator (meta: usage))

```
>Y1 = rand(3,1000); Y2 = rand(3,2000); %generate the data of interest (d=3, T1=1000, T2=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. **Cross-entropy + entropy → Kullback-Leibler divergence**: As is well-known the Kullback-Leibler divergence can be expressed in terms of cross-entropy (see Eq. (96)) and entropy:

$$D(f_1, f_2) = C_{\mathrm{CE}}(f_1, f_2) - H(f_1). \tag{132}$$

   The associated meta divergence estimator is called 'KL_CE_HShannon'.

2. **MMD → energy distance**: As it has been proved recently [86, 140], the energy distance [Eq. (61)] is closely related to MMD [Eq. (52)]:

$$D_{\mathrm{EnDist}}(f_1, f_2) = 2\left[D_{\mathrm{MMD}}(f_1, f_2)\right]^2, \tag{133}$$

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

$$\rho(\mathbf{u}, \mathbf{v}) = k(\mathbf{u}, \mathbf{u}) + k(\mathbf{v}, \mathbf{v}) - 2k(\mathbf{u}, \mathbf{v}). \tag{134}$$

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

3. **Jensen-Shannon divergence**: This divergence is defined [83] in terms of the Shannon entropy as

$$D_{\mathrm{JS}}^{\pi}(f_1, f_2) = H\left(\pi_1\mathbf{y}^1 + \pi_2\mathbf{y}^2\right) - \left[\pi_1 H\left(\mathbf{y}^1\right) + \pi_2 H\left(\mathbf{y}^2\right)\right], \tag{135}$$

   where $\mathbf{y}^i \sim f_i$ and $\pi_1\mathbf{y}^1 + \pi_2\mathbf{y}^2$ denotes the mixture distribution obtained from $\mathbf{y}^1$ and $\mathbf{y}^2$ with π_1, π_2 weights ($\pi_1, \pi_2 > 0$, $\pi_1 + \pi_2 = 1$). The meta estimator is called 'JensenShannon_HShannon' in ITE.

   Notes:

- As it is known $0 \leq D_{\mathrm{JS}}^{\pi}(f_1, f_2) \leq \log(2)$, $D_{\mathrm{JS}}^{\pi}(f_1, f_2) = 0 \Leftrightarrow f_1 = f_2$.
- Specially, for $\pi_1 = \pi_2 = \frac{1}{2}$ we obtain

$$D_{\mathrm{JS}}(f_1, f_2) = D_{\mathrm{JS}}^{\left(\frac{1}{2}, \frac{1}{2}\right)}(f_1, f_2) = H\left(\frac{\mathbf{y}^1 + \mathbf{y}^2}{2}\right) - \frac{H\left(\mathbf{y}^1\right) + H\left(\mathbf{y}^2\right)}{2} = \frac{1}{2}\left[D\left(f_1, \frac{f_1 + f_2}{2}\right) + D\left(f_2, \frac{f_1 + f_2}{2}\right)\right]. \tag{136}$$

  It is known that $\sqrt{D_{\mathrm{JS}}(f_1, f_2)}$ is a (covariant) Hilbertian metric [180, 35, 56]; see Def. 7.
- One can also generalize the Jensen-Shannon divergence [Eq. (135)] to multiple components as

$$D_{\mathrm{JS}}^{\pi}(f_1, \ldots, f_M) = H\left(\sum_{m=1}^{M} \pi_m \mathbf{y}^m\right) - \sum_{m=1}^{M} \pi_m H\left(\mathbf{y}^m\right), \tag{137}$$

  where $\pi_m > 0$ $(m = 1, \ldots, M)$, $\sum_{m=1}^{M} \pi_m = 1$, $\mathbf{y}^m \sim f_m$ $(m = 1, \ldots, M)$.

4. **Jensen-Rényi divergence**: The definition of the Jensen-Rényi divergence is analogous to (137); the difference to D_{JS}^{π} is that the Shannon entropy is changed to the Rényi entropy [52]

$$D_{\mathrm{JR},\alpha}^{\pi}(f_1, \ldots, f_M) = H_{\mathrm{R},\alpha}\left(\sum_{m=1}^{M} \pi_m \mathbf{y}^m\right) - \sum_{m=1}^{M} \pi_m H_{\mathrm{R},\alpha}\left(\mathbf{y}^m\right), \quad (\alpha \geq 0) \tag{138}$$

   where $\pi_m > 0$ $(m = 1, \ldots, M)$, $\sum_{m=1}^{M} \pi_m = 1$, $\mathbf{y}^m \sim f_m$ $(m = 1, \ldots, M)$. The name of the meta estimator is '`JensenRenyi_HRenyi`' in ITE ($M = 2$).

5. **K divergence, L divergence**: The K divergence and the L divergence measures [83] are defined as

$$D_{\mathrm{K}}(f_1, f_2) = D\left(f_1, \frac{f_1 + f_2}{2}\right), \tag{139}$$

$$D_{\mathrm{L}}(f_1, f_2) = D_{\mathrm{K}}(f_1, f_2) + D_{\mathrm{K}}(f_2, f_1). \tag{140}$$

   Notes: They are

   - non-negative, and are zero if and only if $f_1 = f_2$.
   - closely related to the Jensen-Shannon divergence in case of uniform weighting, see Eq. (136).
   - available in ITE ('`K_DKL`' and '`L_DKL`').

6. **Jensen-Tsallis divergence**: The definition of the Jensen-Tsallis divergence [17] follows that of the Jensen-Shannon divergence [Eq. (136)]; only the Shannon entropy is replaced with the Tsallis entropy [Eq. (5)]

$$D_{\mathrm{JT},\alpha}(f_1, f_2) = H_{\mathrm{T},\alpha}\left(\frac{\mathbf{y}^1 + \mathbf{y}^2}{2}\right) - \frac{H_{\mathrm{T},\alpha}\left(\mathbf{y}^1\right) + H_{\mathrm{T},\alpha}\left(\mathbf{y}^2\right)}{2}, \quad (\alpha \neq 1) \tag{141}$$

   where $\mathbf{y}^m \sim f_m$ $(m = 1, 2)$. Notes:

   - The Jensen-Shannon divergence is special case in limit sense:

$$\lim_{\alpha \to 1} D_{\mathrm{JT},\alpha}(f_1, f_2) = D_{\mathrm{JS}}(f_1, f_2). \tag{142}$$

   - The name of the associated meta estimator is '`JensenTsallis_HTsallis`' in ITE.

7. **Symmetric Bregman distance**: This measure [15, 27, 81] can be estimated using Eq. (64); its name is '`symBregman_DBregman`' in ITE.

8. **(Csiszár) f-divergence**: Let us given a convex function f, for which $f(1) = 0$. The f-divergence (also called Csiszár-Morimoto divergence or Ali-Silvey distance) of the probability densities f_1 and f_2 on $\mathbb{R}^d$ is defined [26, 92, 3] as

$$D_f(f_1, f_2) = \int_{\mathbb{R}^d} f\left[\frac{f_1(\mathbf{u})}{f_2(\mathbf{u})}\right] f_2(\mathbf{u}) \mathrm{d}\mathbf{u}. \tag{143}$$

   Notes:

Estimated quantity	Principle	d	`cost_name`
J-distance (D_{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_{EnDist})	pairwise distances, equivalence to MMD	$d \geq 1$	'EnergyDist_DMMD'
Jensen-Shannon divergence (D^{π}_{JS})	smoothed (π), defined via the Shannon entropy	$d \geq 1$	'JensenShannon_HShannon'
Jensen-Rényi divergence ($D^{\pi}_{\mathrm{JR},\alpha}$)	smoothed (π), defined via the Rényi entropy	$d \geq 1$	'JensenRenyi_HRenyi'
K divergence (D_{K})	smoothed Kullback-Leibler divergence	$d \geq 1$	'K_DKL'
L divergence (D_{L})	symmetrised K divergence	$d \geq 1$	'L_DKL'
Jensen-Tsallis divergence ($D_{\mathrm{JT},\alpha}$)	smoothed, defined via the Tsallis entropy	$d \geq 1$	'JensenTsallis_HTsallis'
Symmetric Bregman distance ($D_{\mathrm{SB},\alpha}$)	symmetrised Bregman distance	$d \geq 1$	'symBregman_DBregman'
Maximum mean discrepancy (D_{MMD})	block-average of U-statistic based MMDs	$d \geq 1$	'BMMD_DMMD_Ustat'
f-divergence (D_{f})	second-order Taylor expansion + χ^2 divergence	$d \geq 1$	'f_DChiSquare'

Table 10: Divergence estimators (meta). Third column: dimension (d) constraint.

- $D_f(f_1, f_2) \geq 0$ with equality if and only if $f_1 = f_2$.
- Using a second-order Taylor expansion of f around 1, the f-divergence can be approximated [8, 104] using the Pearson χ^2 divergence [see Eq. (68)]

$$D_f(f_1, f_2) \approx \frac{f''(1)}{2} D_{\chi^2}(f_1, f_2). \tag{144}$$

The calling form the meta divergence estimators is unified, one only has to change in Example 18 the `cost_name` to the value in the last column of Table 10.

3.2.4 Association Measure Estimators

One can define and use meta association measure estimators completely analogously to meta mutual information estimators (see Section 3.2.2). The meta association measure estimators included in ITE are the

- **Correntropy induced metric** [84, 141], **centered correntropy induced metric** [128]:

$$A_{\mathrm{CIM}}\left(y^1, y^2\right) = \sqrt{k(0,0) - A_{\mathrm{CorrEntr}}\left(y^1, y^2\right)}, \tag{145}$$

$$A_{\mathrm{CCIM}}\left(y^1, y^2\right) = \sqrt{A_{\mathrm{CCorrEntr}}\left(y^1, y^1\right) + A_{\mathrm{CCorrEntr}}\left(y^2, y^2\right) - 2A_{\mathrm{CCorrEntr}}\left(y^1, y^2\right)}, \tag{146}$$

  where k is the kernel used in the correntropy estimator [Eqs. (85)-(86)]. The corresponding meta estimators are called '`CIM`' and '`CCIM`' in ITE.

- **Lower tail dependence via conditional Spearman's ρ**: This lower tail dependence measure has been defined [134] as the limit of $\hat{A}_{\rho_{\mathrm{lt}}} = \hat{A}_{\rho_{\mathrm{lt}}}(p)$ [Eq. (94)]:

$$A_{\rho_{\mathrm{L}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\mathrm{L}}}(C) = \lim_{p \to 0, p > 0} A_{\rho_{\mathrm{lt}}}(C) = \lim_{p \to 0, p > 0} \frac{d+1}{p^{d+1}} \int_{[0,p]^d} C(\mathbf{u}) \mathrm{d}\mathbf{u}, \tag{147}$$

  provided that the limit exists. The name of the association measure is '`Spearman_L`' in ITE.

  Note:

  - Similarly to $A_{\rho_{\mathrm{lt}}}$ [Eq. (94)], $A_{\rho_{\mathrm{L}}}$ preserves concordance ordering [see Eq. (76)]: $C_1 \prec C_2 \Rightarrow A_{\rho_{\mathrm{L}}}(C_1) \leq A_{\rho_{\mathrm{L}}}(C_2)$.
  - Moreover, $0 \leq A_{\rho_{\mathrm{L}}}(C) \leq 1$; the comomonotonic copula M implies $A_{\rho_{\mathrm{L}}} = 1$ and the independence copula Π yields $A_{\rho_{\mathrm{L}}} = 0$.
  - $A_{\rho_{\mathrm{L}}}$ can be used as an alternative of the tail-dependence coefficient [146] widely spreaded in bivariate extreme value theory:

$$\lambda_L = \lambda_L(C) = \lim_{p \to 0, p > 0} \frac{C(p,p)}{p}. \tag{148}$$

    An important drawback of λ_L is that it takes into account the copula only on the diagonal ($C(p,p)$).

Estimated quantity	Principle	d_m	M	`cost_name`
correntropy induced metric (A_{CIM})	metric from correntropy	$d_m = 1$	$M = 2$	`'CIM'`
centered correntropy induced metric (A_{CCIM})	metric from centered correntropy	$d_m = 1$	$M = 2$	`'CCIM'`
lower tail dependence via conditional Spearman's ρ ($A_{\rho_{\mathrm{L}}}$)	limit of $A_{\rho_{\mathrm{lt}}}$	$d_m = 1$	$M \geq 2$	`'Spearman_L'`
upper tail dependence via conditional Spearman's ρ ($A_{\rho_{\mathrm{U}}}$)	limit of $A_{\rho_{\mathrm{ut}}}$	$d_m = 1$	$M \geq 2$	`'Spearman_U'`

Table 11: Association measure 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; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$).

- **Upper tail dependence via conditional Spearman's ρ**: This upper tail dependence measure has been introduced in [134] as the limit of $\hat{A}_{\rho_{\mathrm{ut}}} = \hat{A}_{\rho_{\mathrm{ut}}}(p)$ [Eq. (95)]:

$$A_{\rho_{\mathrm{U}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\mathrm{U}}}(C) = \lim_{p \to 0, p > 0} A_{\rho_{\mathrm{ut}}}(C), \tag{149}$$

  provided that the limit exists. The measure is an analogue of (147) in the 'upper' domain. It is called '`Spearman_U`' in ITE.

The meta association measure estimators are summarized in Table 11.

3.2.5 Cross Quantity Estimators

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

3.2.6 Estimators of Kernels on Distributions

It is possible to define and use meta estimators of kernels on distributions similarly to meta divergence estimators (see Section 3.2.3). ITE contains the following meta estimators (see Table 12):

Jensen-Shannon kernel: The Jensen-Shannon kernel [87, 88] is defined as

$$K_{\mathrm{JS}}\left(f_1, f_2\right) = \log(2) - D_{\mathrm{JS}}(f_1, f_2), \tag{150}$$

where D_{JS} is the Jensen-Shannon divergence [Eq. (136)]. The corresponding meta estimator is called '`JS_DJS`' in ITE.

Jensen-Tsallis kernel: The definition of the Jensen-Tsallis kernel [88] is similar to that of the Jensen-Shannon kernel [see Eq. (150)]

$$K_{\mathrm{JT},\alpha}\left(f_1, f_2\right) = \log_\alpha(2) - T_\alpha(f_1, f_2), \quad (\alpha \in [0,2] \backslash \{1\}) \tag{151}$$

$$\log_\alpha(x) = \frac{x^{1-\alpha} - 1}{1 - \alpha}, \tag{152}$$

$$T_\alpha(f_1, f_2) = H_{\mathrm{T},\alpha}\left(\frac{\mathbf{y}^1 + \mathbf{y}^2}{2}\right) - \frac{H_{\mathrm{T},\alpha}\left(\mathbf{y}^1\right) + H_{\mathrm{T},\alpha}\left(\mathbf{y}^2\right)}{2^\alpha}, \tag{153}$$

where $\mathbf{y}^m \sim f_m$ $(m = 1, 2)$, $\log_\alpha$ is the α-logarithm function [181], $H_{\mathrm{T},\alpha}$ is the Tsallis entropy [see Eq. (5)], and $T_\alpha(f_1, f_2)$ is the so-called Jensen-Tsallis α-difference of f_1 and f_2.

Notes:

- The Jensen-Shannon kernel is a special case in the limit

$$\lim_{\alpha \to 1} K_{\mathrm{JT},\alpha}\left(f_1, f_2\right) = K_{\mathrm{JS}}\left(f_1, f_2\right). \tag{154}$$

- The meta estimator is called '`JT_HJT`' in ITE.

Exponentiated Jensen-Shannon kernel: The exponentiated Jensen-Shannon kernel [87, 88] is defined as

$$K_{\mathrm{EJS},u}(f_1,f_2) = e^{-uD_{\mathrm{JS}}(f_1,f_2)}, \quad (u>0) \tag{155}$$

where D_{JS} is the Jensen-Shannon divergence [Eq. (136)]. The corresponding meta estimator is called '`EJS_DJS`' in ITE.

Exponentiated Jensen-Rényi kernel(s): The exponentiated Jensen-Rényi kernels [88] are defined as

$$K_{\mathrm{EJR1},u,\alpha}(f_1,f_2) = e^{-uH_{\mathrm{R},\alpha}\left(\frac{\mathbf{y}^1+\mathbf{y}^2}{2}\right)}, \quad (u>0,\ \alpha\in[0,1)) \tag{156}$$

$$K_{\mathrm{EJR2},u,\alpha}(f_1,f_2) = e^{-uD_{\mathrm{JR},\alpha}(f_1,f_2)}, \quad (u>0,\ \alpha\in[0,1)) \tag{157}$$

where $\mathbf{y}^m \sim f_m$ $(m=1,2)$, $H_{\mathrm{R},\alpha}$ is the Rényi entropy [see Eq. (3)], and the $D_{\mathrm{JR},\alpha}$ Jensen-Rényi divergence is the uniformly weighted special case of (138):

$$D_{\mathrm{JR},\alpha}(f_1,f_2) = D_{\mathrm{JR},\alpha}^{\left(\frac{1}{2},\frac{1}{2}\right)}(f_1,f_2) = H_{\mathrm{R},\alpha}\left(\frac{\mathbf{y}^1+\mathbf{y}^2}{2}\right) - \frac{H_{\mathrm{R},\alpha}\left(\mathbf{y}^1\right)+H_{\mathrm{R},\alpha}\left(\mathbf{y}^2\right)}{2}. \tag{158}$$

Notes:

- The exponentiated Jensen-Shannon kernel is a special case in limit sense

$$\lim_{\alpha\to 1} K_{\mathrm{EJR2},u,\alpha}(f_1,f_2) = K_{\mathrm{EJS},u}(f_1,f_2). \tag{159}$$

- The corresponding meta estimators are called '`EJR1_HR`', '`EJR2_DJR`'.

Exponentiated Jensen-Tsallis kernel(s): The exponentiated Jensen-Tsallis kernels [88] are defined as

$$K_{\mathrm{EJT1},u,\alpha}(f_1,f_2) = e^{-uH_{\mathrm{T},\alpha}\left(\frac{\mathbf{y}^1+\mathbf{y}^2}{2}\right)}, \quad (u>0, \alpha\in[0,2]\backslash\{1\}) \tag{160}$$

$$K_{\mathrm{EJT2},u,\alpha}(f_1,f_2) = e^{-uD_{\mathrm{JT},\alpha}(f_1,f_2)}, \quad (u>0, \alpha\in[0,2]\backslash\{1\}) \tag{161}$$

where $\mathbf{y}^m \sim f_m$ $(m=1,2)$, $H_{\mathrm{T},\alpha}$ is the Tsallis entropy [see Eq. (5)], and $D_{\mathrm{T},\alpha}$ is the Jensen-Tsallis divergence [see Eq. (141)].

Notes:

- The exponentiated Jensen-Shannon kernel is a special case in limit sense

$$\lim_{\alpha\to 1} K_{\mathrm{EJT2},u,\alpha}(f_1,f_2) = K_{\mathrm{EJS},u}(f_1,f_2). \tag{162}$$

- The corresponding meta estimators are called '`EJT1_HT`', '`EJT2_DJT`'.

Let us take a simple estimation example:

Example 19 (Kernel estimation on distributions (meta: 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 = KJS_DJS_initialization(mult); %initialize the kernel ('K') estimator on distributions ('JS_DJS')
>K = KJS_DJS_estimation(Y1,Y2,co);  %perform kernel estimation on distributions
```

3.3 Uniform Syntax of the Estimators, List of Estimated Unconditional Quantities

The modularity of the ITE package in terms of (i) the definition and usage of the estimators of base/meta entropy, mutual information, divergence, association measures, cross quantities, kernels on distributions, and (ii) the possibility to simple embed novel estimators can be assured by following the templates given in Section 3.3.1 and Section 3.3.2. The default templates are detailed in Section 3.3.1. Section 3.3.2 is about user-specified parameters, inheritance in meta estimators and a list of the estimated unconditional quantities (Fig. 1).

Estimated quantity	Principle	d	cost_name
Jensen-Shannon kernel (K_{JS})	function of the Jensen-Shannon divergence	$d \geq 1$	'JS_DJS'
Jensen-Tsallis kernel ($K_{\mathrm{JT},\alpha}$)	function of the Tsallis entropy	$d \geq 1$	'JT_HJT'
exponentiated Jensen-Shannon kernel ($K_{\mathrm{EJS},u}$)	function of the Jensen-Shannon divergence	$d \geq 1$	'EJS_DJS'
exponentiated Jensen-Rényi kernel-1 ($K_{\mathrm{EJR1},u,\alpha}$)	function of the Rényi entropy	$d \geq 1$	'EJR1_HR'
exponentiated Jensen-Rényi kernel-2 ($K_{\mathrm{EJR2},u,\alpha}$)	function of the Jensen-Rényi divergence	$d \geq 1$	'EJR2_DJR'
exponentiated Jensen-Tsallis kernel-1 ($K_{\mathrm{EJT1},u,\alpha}$)	function of the Tsallis entropy	$d \geq 1$	'EJT1_HT'
exponentiated Jensen-Tsallis kernel-2 ($K_{\mathrm{EJT2},u,\alpha}$)	function of the Jensen-Tsallis divergence	$d \geq 1$	'EJT2_DJT'

Table 12: Estimators of kernels on distributions (meta). Third column: dimension (d) constraint.

3.3.1 Default Usage

In this section the templates of the unconditional information theoretical estimators are enlisted:

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 measure estimator: initialization)

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

Template 5 (Cross quantity estimator: initialization)

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

Template 6 (Kernel on distributions: initialization)

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

2. Estimation:

Template 7 (Entropy estimator: estimation)

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

Template 8 (Mutual information estimator: estimation)

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

Template 9 (Divergence estimator: estimation)

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

Template 10 (Association measure estimator: estimation)

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

Template 11 (Cross quantity estimator: estimation)

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

Template 12 (Kernel on distributions: estimation)

```
function [K] = K<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`, `K_initialization.m`,
- for estimation: `H_estimation.m`, `I_estimation.m`, `D_estimation.m`, `A_estimation.m`, `C_estimation.m`, `K_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 [co] = K_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)
function [K] = K_estimation(Y1,Y2,co)
```

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

- entropy: from the last column of Table 2 and Table 8,
- mutual information: from the last column of Table 3 and Table 9,
- divergence: from the last column of Table 4 and Table 10,
- association measures: from the last column of Table 5,
- cross quantities: from the last column of Table 6.
- kernels on distributions: from the last column of Table 7.

By the ITE construction, following for

- entropy: Template 1 (initialization) and Template 7 (estimation),
- mutual information: Template 2 (initialization) and Template 8 (estimation),
- divergence: Template 3 (initialization) and Template 9 (estimation),
- association measure: Template 4 (initialization) and Template 10 (estimation),
- cross quantity: Template 5 (initialization) and Template 11 (estimation),
- kernel on distributions: Template 6 (initialization) and Template 12 (estimation),

user-defined estimators can be immediately used. Let us demonstrate idea of the high-level initialization and estimation with a simple example, Example 3 can equivalently be written as:[21]

Example 20 (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 5. 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 1-9) 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.

[21] One can perform mutual information, divergence, association measure, cross quantity and distribution kernel estimations similarly.

3.3.2 User-defined Parameters; Inheritence in Meta Estimators

Beyond the syntax of Section 3.3.1, ITE supports an alternative initialization solution (provided that the estimator has parameters)

`function [co] = <H/I/D/A/C/K><cost_name>_initialization(mult,post_init)`

where `post_init` is a {field_name1,field_value1,field_name2,field_value2,...} cell array containing the user-defined field (name,value) pairs. The construction has two immediate applications:

User-specified parameters: The user can specify (and hence override) the default field values of the estimators. As an example, let us take the Rényi entropy estimator using k-nearest neighbors (`Renyi_kNN_k`).

Example 21 (User-specified field values (overriding the defaults))

```
>mult = 1; %multiplicative constant is important
>co = HRenyi_kNN_k_initialization(mult,{'alpha',0.9});       %set alpha
>co = HRenyi_kNN_k_initialization(mult,{'alpha',0.9,'k',4}); %set alpha and the number of
                                                             %nearest neighbors
>co = HRenyi_kNN_k_initialization(mult,{'kNNmethod','ANN','k',3,'epsi',0}); %set the nearest
                                                            %neighbor method used and its parameters
```

The same solution can be used in the high-level mode:

Example 22 (User-specified field values (overriding the defaults; high-level))

```
>mult = 1; %multiplicative constant is important
>cost_name = 'Renyi_kNN_k'; %cost name
>co = H_initialization(cost_name,mult,{'alpha',0.9});       %set alpha
>co = H_initialization(cost_name,mult,{'alpha',0.9,'k',4}); %set alpha and the number of
                                                           %nearest neighbors
>co = H_initialization(cost_name,mult,{'kNNmethod','ANN','k',3,'epsi',0}); %set the nearest
                                                          %neighbor method used and its parameters
```

Inheritance in meta estimators: In case of meta estimators, inheritence of certain parameter values may be desirable. As an example let us take the Rényi mutual information estimator based on the Rényi divergence (`Renyi_DRenyi`, Eq. (115)). Here, the α parameter of the Rényi divergence is set to the one used in Rényi mutual information (see `IRenyi_DRenyi_initialization.m`; only the relevant part is detailed):

Example 23 (Inheritance in meta estimators)

```
function [co] = IRenyi_DRenyi_initialization(mult,post_init)
...
co.alpha = 0.99; %alpha parameter
co.member_name = 'Renyi_kNN_k'; %name of the Rényi divergence estimator used
...
co.member_co = D_initialization(co.member_name,mult,{'alpha',co.alpha});%the 'alpha' field
                                                                        %(co.alpha) is inherited
```

The meta estimators using this technique are enlisted in Table 13. All the unconditional ITE estimators are listed in Fig. 1.

3.4 Guidance on Estimator Choice

It is difficult (if not impossible) to provide a general guidance on the choice of the estimator to be used. The applicability of an estimator depends highly on the

1. application considered,

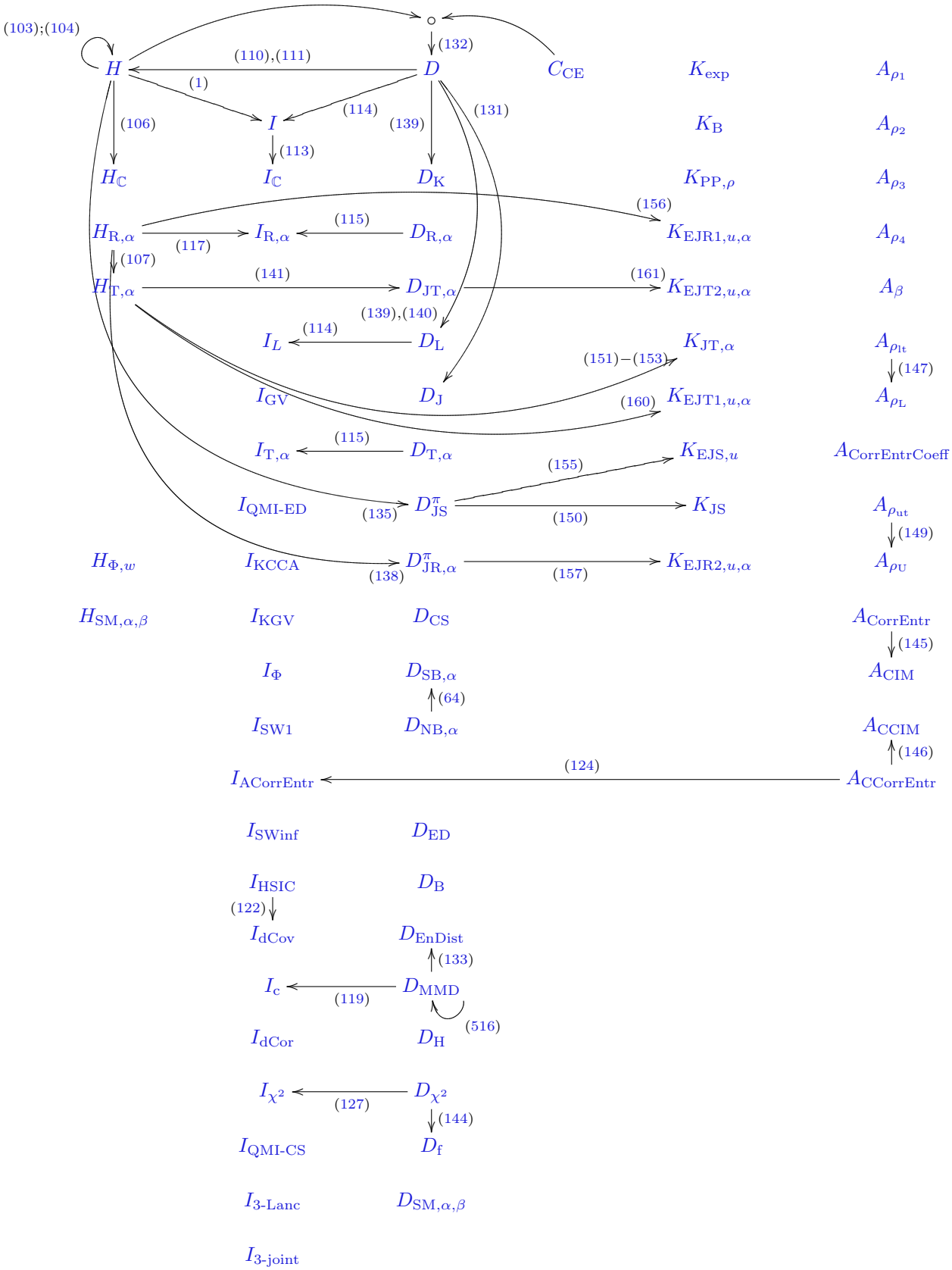


Figure 1: List of the estimated unconditional information theoretical quantities. Columns from left to right: entropy, mutual information, divergence, cross quantities, kernels on distributions, association measures. '$X \xrightarrow{Z} Y$' means: The Y quantity can be estimated in ITE by a meta method from quantity X using Eq. (Z). Click on the quantities to see their definitions.

Inherited parameter			Equation	Function
$H_{\mathrm{T},\alpha}$	$\xrightarrow{\alpha}$	$H_{\mathrm{R},\alpha}$	(107)	`HTsallis_HRenyi_initialization.m`
$I_{\mathrm{R},\alpha}$	$\xrightarrow{\alpha}$	$D_{\mathrm{R},\alpha}$	(115)	`IRenyi_DRenyi_initialization.m`
$I_{\mathrm{T},\alpha}$	$\xrightarrow{\alpha}$	$D_{\mathrm{T},\alpha}$	(115)	`ITsallis_DTsallis_initialization.m`
$I_{\mathrm{R},\alpha}$	$\xrightarrow{\alpha}$	$H_{\mathrm{R},\alpha}$	(117)	`IRenyi_HRenyi_initialization.m`
$D^{\pi}_{\mathrm{JR},\alpha}$	$\xrightarrow{\alpha}$	$H_{\mathrm{R},\alpha}$	(138)	`DJensenRenyi_HRenyi_initialization.m`
$D_{\mathrm{JT},\alpha}$	$\xrightarrow{\alpha}$	$H_{\mathrm{T},\alpha}$	(141)	`DJensenTsallis_HTsallis_initialization.m`
$D_{\mathrm{SB},\alpha}$	$\xrightarrow{\alpha}$	$D_{\mathrm{NB},\alpha}$	(64)	`DsymBregman_DBregman_initialization.m`
K_{JS}	$\xrightarrow{\pi=\left[\frac{1}{2};\frac{1}{2}\right]}$	D^{π}_{JS}	(150)	`KJS_DJS_initialization.m`
$K_{\mathrm{JT},\alpha}$	$\xrightarrow{\alpha}$	$H_{\mathrm{T},\alpha}$	(151)-(153)	`KJT_HJT_initialization.m`
$K_{\mathrm{EJS},u}$	$\xrightarrow{\pi=\left[\frac{1}{2};\frac{1}{2}\right]}$	D^{π}_{JS}	(155)	`KEJS_DJS_initialization.m`
$K_{\mathrm{EJR1},u,\alpha}$	$\xrightarrow{\alpha}$	$H_{\mathrm{R},\alpha}$	(156)	`KEJR1_HR_initialization.m`
$K_{\mathrm{EJR2},u,\alpha}$	$\xrightarrow{\pi=\left[\frac{1}{2};\frac{1}{2}\right],\alpha}$	$D^{\pi}_{\mathrm{JR},\alpha}$	(157)	`KEJR2_DJR_initialization.m`
$K_{\mathrm{EJT1},u,\alpha}$	$\xrightarrow{\alpha}$	$H_{\mathrm{T},\alpha}$	(160)	`KEJT1_HT_initialization.m`
$K_{\mathrm{EJT2},u,\alpha}$	$\xrightarrow{\alpha}$	$D_{\mathrm{JT},\alpha}$	(161)	`KEJT2_DJT_initialization.m`

Table 13: Inheritence in meta estimators. Notation $X \xrightarrow{z} Y$: the X meta method sets the z parameter(s) of the Y estimator. Second column: definition of the meta estimator.

2. importance of theoretical guarantees,
3. computational resources/bottlenecks, scaling requirements.

Such a guidance is also somewhat subjective – and hence may vary from person to person. At the same time, the author of the ITE toolbox feels it important to help the users of the ITE package – and highly encourage them to explore the advantages/disadvantages of the estimators, the 'best' method for their particular application considered. Below some personal experiences/guidelines are enlisted:

1. *Non plug-in* type estimators generally scale favourable in dimension compared to their plug-in type counterparts; since there is no need to estimate densities 'only' functionals of the densities.
2. In one dimension ($d = 1$), *spacing* based entropy estimators provide fast estimators with sound theoretical guarantees.
3. *kNN methods* can be considered as the direct extensions of spacing solutions to the multidimensional case with consistency results. In case of multidimensional entropy estimation problems, the Rényi entropy can be a practical first choice: (i) it can be estimated via kNN methods and (ii) under special parameterization ($\alpha \to 1$) it covers the well-known Shannon entropy.
4. A further relevant property of kNN techniques is that they can be applied for the consistent estimation of other information theoretical quantities including numerous mutual information, divergence measures, and cross-entropy.
5. kNN methods can be somewhat sensitive to *outliers*; this can be alleviated, for example by *minimum spanning tree* techniques with higher computational burden.
6. An alternative very successful direction to cope with outliers, is the *copula technique* that relies on order statistics of the data.
7. *Kernel methods* represent one of the most successful directions in dependence estimation, e.g., in blind signal separation problems:
   (a) by the kernel trick they make it possible to compute measures defined over infinite dimensonal function spaces,
   (b) their estimation often reduces to well-studied subtasks, such as (generalized) eigenvalue problems, fast pairwise distance computations.
8. Using *smoothed* divergence measures, requirements of the absolute continuity of the underlying measures (one to the other) can be circumvented.

Estimated quantity	Principle	d_m	`cost_name`
conditional Shannon entropy [$H(\cdot\|\cdot)$]	reduction to Shannon entropy	$d_m \geq 1$	`'Shannon_HShannon'`

Table 14: Conditional entropy estimators (meta). Third column: dimension (d_m) constraint.

4 Estimation of Conditional Quantities

This section is about conditional information theoretical estimators. The quantities follow the same base/meta naming conventions as their unconditional counterparts (Section 3). The conditional ITE estimators are listed in Fig. 2.

4.1 Meta Estimators

Below we enlist the conditional meta-estimators of ITE.

4.1.1 Conditional Entropy Estimators

1. **Conditional Shannon entropy**: It is defined as

$$\begin{aligned} H\left(\mathbf{y}^1|\mathbf{y}^2\right) &= \mathbb{E}_{\mathbf{y}^2}\left[H\left(\mathbf{y}^1|\mathbf{y}^2\right)\right] \\ &= H\left(\left[\mathbf{y}^1;\mathbf{y}^2\right]\right) - H\left(\mathbf{y}^2\right). \end{aligned} \tag{163}$$

   The name of the associated estimator is '`Shannon_HShannon`'.

The calling syntax of the conditional entropy estimator is unified in ITE, for example estimation of Eq. (163) can be carried out as:

Example 24 (Conditional entropy estimation (meta: usage))

```
>d1 = 2; d2 = 1;                          %dimensions
>Y = rand(d1+d2,5000);                    %generate samples from [y^1;y^2] (T=5000)
>Y1 = Y(1:d1,:); Y2 = Y(d1+1:end,:);      %extract Y1 and Y2
>mult = 1;                                %multiplicative constant is important
>co = condHShannon_HShannon_initialization(mult);     %initialize the conditional entropy ('condH')
                                                      %estimator ('Shannon_HShannon')
>condH = condHShannon_HShannon_estimation(Y1,Y2,co);  %perform estimation
```

The conditional entropy estimators are enlisted in Table 14.

4.1.2 Conditional Mutual Information Estimators

1. **Conditional Shannon mutual information**: The quantity is defined as

$$\begin{aligned} I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M|\mathbf{y}^{M+1}\right) &= \mathbb{E}_{\mathbf{y}^{M+1}}\left[I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M|\mathbf{y}^{M+1}\right)\right] \\ &= -H\left(\left[\mathbf{y}^1;\ldots;\mathbf{y}^{M+1}\right]\right) + \sum_{m=1}^{M} H\left(\left[\mathbf{y}^m;\mathbf{y}^{M+1}\right]\right) - (M-1)H\left(\mathbf{y}^{M+1}\right). \end{aligned} \tag{164}$$

   The corresponding estimator is called '`Shannon_HShannon`'.

The calling syntax of the conditional mutual information estimator is unified in ITE, for example to estimate Eq. (164):

Example 25 (Conditional mutual information estimation (meta: usage))

```
>ds = [2;1;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 = condIShannon_HShannon_initialization(mult);   %initialize the conditional mutual
                                                    %information ('condI') estimator ('Shannon_HShannon')
>condI = condIShannon_HShannon_estimation(Y,ds,co); %perform estimation
```

Estimated quantity	Principle	d_m	M	`cost_name`
conditional Shannon mutual information [$I(\cdot\|\cdot)$]	reduction to Shannon entropy	$d_m \geq 1$	$M \geq 2$	`'Shannon_HShannon'`

Table 15: Conditional 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; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M; \mathbf{y}^{M+1}\right]$)..

The conditional mutual information estimators are enlisted in Table 14.

4.2 Templates, List of Estimated Conditional Quantities

In this section the templates of the conditional information theoretical estimators are enlisted:

1. Initialization:

   Template 13 (Conditional entropy estimator: initialization)

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

   Template 14 (Conditional mutual information estimator: initialization)

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

2. Estimation:

   Template 15 (Conditional entropy estimator: estimation)

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

   Template 16 (Conditional mutual information estimator: estimation)

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

The unified design of ITE enables high-level initialization and estimation: functions are

- for initialization: `condH_initialization.m`, `condI_initialization.m`,
- for estimation: `condH_estimation.m`, `condI_estimation.m`,

following the templates:

```
function [co] = condH_initialization(cost_name,mult)
function [co] = condI_initialization(cost_name,mult)

function [condH] = condH_estimation(Y1,Y2,co)
function [condI] = condI_estimation(Y,ds,co)
```

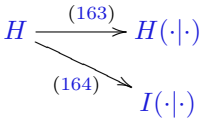


Figure 2: List of the estimated conditional information theoretical quantities. '$X \xrightarrow{Z} Y$' means: The Y quantity can be estimated in ITE by a meta method from quantity X using Eq. (Z). Click on the quantities to see their definitions.

Here, the `cost_name` of the entropy, mutual information can come from

- entropy: from the last column of Table 14,
- mutual information: from the last column of Table 15.

Let us take a high-level initialization and estimation example corresponding to Example 24:

Example 26 (Conditional entropy estimation (high-level, usage))

```
>d1 = 2; d2 = 1;                          %dimensions
>Y = rand(d1+d2,5000);                    %generate samples from [y^1;y^2] (T=5000)
>Y1 = Y(1:d1,:); Y2 = Y(d1+1:end,:);      %extract Y1 and Y2
>mult = 1;                                %multiplicative constant is important
>cost_name = 'Shannon_HShannon';          %name of the estimator
>co = condH_initialization(cost_name,mult); %initialize the conditional entropy estimator
>condH = condH_estimation(Y1,Y2,co);       %perform estimation
```

5 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 5.1 formulates the problem domain, the independent process analysis (IPA) problem family. In Section 5.2 the solution methods of IPA are detailed. Section 5.3 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 5.4.

5.1 IPA Models

In Section 5.1.1 we focus on the simplest linear model, which allows hidden, independent multidimensional sources (subspaces), the so-called independent subspace analysis (ISA) problem. Section 5.1.2 is about the extensions of ISA.

5.1.1 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.

The ISA equations One may think of independent subspace analysis (ISA)[22] [18, 30] 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 recordings (observations) only.

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

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,

$$\mathbf{x}_t = \mathbf{A}\mathbf{e}_t, \tag{165}$$

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 [69, 20, 22] 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) [59]. 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 ($\mathbf{y}^m$),

$$J_{\mathrm{I}}(\mathbf{W}) = I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) \rightarrow \min_{\mathbf{W} \in GL(D)}, \tag{166}$$

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. (166)] can also be expressed from only *pair-wise* mutual information by recursive methods [25]

$$I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = \sum_{m=1}^{M-1} I\left(\mathbf{y}^m, \left[\mathbf{y}^{m+1}, ..., \mathbf{y}^M\right]\right). \tag{167}$$

Thus, an equivalent information theoretical ISA objective to (166) is

$$J_{\mathrm{Irecursive}}(\mathbf{W}) = \sum_{m=1}^{M-1} I\left(\mathbf{y}^m, \left[\mathbf{y}^{m+1}, ..., \mathbf{y}^M\right]\right) \rightarrow \min_{\mathbf{W} \in GL(D)}. \tag{168}$$

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 $\mathbf{A}^*\mathbf{A} = \mathbf{I}$, and
- the task is complete ($D = D_x = D_e$),

one can restrict the optimization in (166) and (168) to the orthogonal group ($\mathbf{W} \in \mathcal{O}^D$). Under the whiteness assumption, well-known identities of mutual information and entropy expressions [25] show that the ISA problem is equivalent to

$$J_{\mathrm{sumH}}(\mathbf{W}) = \sum_{m=1}^{M} H\left(\mathbf{y}^m\right) \rightarrow \min_{\mathbf{W} \in \mathcal{O}^D}, \tag{169}$$

$$J_{H,I}(\mathbf{W}) = \sum_{m=1}^{M} \sum_{i=1}^{d_m} H(y_i^m) - \sum_{m=1}^{M} I\left(y_1^m, \ldots, y_{d_m}^m\right) \rightarrow \min_{\mathbf{W} \in \mathcal{O}^D}, \tag{170}$$

$$J_{I,I}(\mathbf{W}) = I\left(y_1^1, \ldots, y_{d_M}^M\right) - \sum_{m=1}^{M} I\left(y_1^m, \ldots, y_{d_m}^m\right) \rightarrow \min_{\mathbf{W} \in \mathcal{O}^D}, \tag{171}$$

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[23] within the subspaces [179].

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 [18]: 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 [167]. 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 [170]. The addressed extension directions are (i) presented in Section 5.1.2, (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

$$\mathbf{W}_{\text{ISA}} = \mathbf{P}\mathbf{W}_{\text{ICA}}, \tag{172}$$

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 (170) and (171)], the ISA problem is equivalent to

$$J_{\text{I}}(\mathbf{P}) = I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}, \tag{173}$$

$$J_{\text{Irecursive}}(\mathbf{P}) = \sum_{m=1}^{M-1} I\left(\mathbf{y}^m, \left[\mathbf{y}^{m+1}, ..., \mathbf{y}^M\right]\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}, \tag{174}$$

$$J_{\text{sumH}}(\mathbf{P}) = \sum_{m=1}^{M} H\left(\mathbf{y}^m\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}, \tag{175}$$

$$J_{\text{sum-I}}(\mathbf{P}) = -\sum_{m=1}^{M} I\left(y_1^m, ..., y_{d_m}^M\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}. \tag{176}$$

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 [22]. However, minimization of the pairwise dependence of the estimated subspaces

$$J_{\text{Ipairwise}}(\mathbf{P}) = \sum_{m_1\neq m_2} I\left(\mathbf{y}^{m_1}, \mathbf{y}^{m_2}\right) \to \min_{\mathbf{P}\in\mathcal{P}^D} \tag{177}$$

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

$$J_{\text{Ipairwise1d}}(\mathbf{P}) = \sum_{m_1,m_2=1;m_1\neq m_2}^{M} \sum_{i_1=1}^{d_{m_1}} \sum_{i_2=1}^{d_{m_2}} I\left(y_{i_1}^{m_1}, y_{i_2}^{m_2}\right) \to \min_{\mathbf{P}\in\mathcal{P}^D}. \tag{178}$$

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

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

ISA optimization methods Let us fix an ISA objective J [Eq. (173)-(178)]. 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., (176)), 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 [160].

'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 [166]: the cross-entropy solution suggested for the traveling salesman problem [132] 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

$$D = d_1 + \ldots + d_M \quad (d_m > 0, M \leq D) \tag{179}$$

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 [53, 183]:

$$f(D) \sim \frac{e^{\pi\sqrt{2D/3}}}{4D\sqrt{3}} \tag{180}$$

as $D \to \infty$. An efficient method with good scaling properties has been put forth in [121] for searching the permutation group for the ISA separation theorem (see Table 16). 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*. The approach can be considered as objective (178) 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 [195].

5.1.2 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 5.3). The ITE package directly implements the estimation of the following models[24] (the relations of the different models are summarized in Fig.3):

Linear systems:

[24]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'.

AR-IPA:

Equations, assumptions: In the AR-IPA (autoregressive-IPA) task [62] ($d_m = 1$, $\forall m$), [123] ($d_m \geq 1$), 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 $\mathbf{s}$:

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad \mathbf{s}_t = \sum_{i=1}^{L_s} \mathbf{F}_i \mathbf{s}_{t-i} + \mathbf{e}_t, \tag{181}$$

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. (181) can be rewritten in the following concise form:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \qquad \mathbf{F}[z]\mathbf{s} = \mathbf{e} \tag{182}$$

using the polynomial of the time-shift operator $\mathbf{F}[z] := \mathbf{I} - \sum_{i=1}^{L_s} \mathbf{F}_i z^i \in \mathbb{R}[z]^{D\times D}$ [78]. 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 ($\mathbf{A} \in GL(D)$), and
3. $\mathbf{e}$ satisfies the ISA assumptions (see Section 5.1.1)

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 $\mathbf{x}$ [62, 123]. 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) [167]. 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 ($\mathbf{x} \in \mathbb{R}^{D_x}$)

$$\mathbf{x}_t = \sum_{l=0}^{L_e} \mathbf{H}_l \mathbf{e}_{t-l}, \tag{183}$$

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[25], 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 [113] $d_m = 1$ ($\forall m$), and L_e is a non-negative integer.

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 [167]. 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 [168]. Demo: `demo_uMA_IPA_LPA.m`.

[25] One can show for $D_x > D_e$ that under mild conditions $\mathbf{H}[z]$ has a left inverse with probability 1 [126]; e.g., when the matrix $[\mathbf{H}_0, \ldots, \mathbf{H}_{L_e}]$ is drawn from a continuous distribution.

- 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(∞)] process, one [156] can construct an efficient estimation method for the hidden components via an asymptotically consistent LPA procedure augmented with ISA. Such AR(∞) representation can be guaranteed by assuming the stability of $\mathbf{H}[z]$ [41]. Demo: `demo_MA_IPA_LPA.m`.

Post nonlinear models:

Equations, assumptions: In the post nonlinear ISA (PNL-ISA) problem [171] 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 ($\mathbf{e} \in \mathbb{R}^D$):

$$\mathbf{x}_t = \mathbf{g}(\mathbf{A}\mathbf{e}_t), \tag{184}$$

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 [176] (for a review see [70]), 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 [171]. 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. (105), (106), (113)]. 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 ($\mathbf{e}_t$):

$$\mathbf{x}_t = \mathbf{A}\mathbf{e}_t, \tag{185}$$

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, and
- $\mathbf{e}^m \in \mathbb{C}^{d_m}$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 an 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 [162]. Demo: `demo_complex_ISA.m`.
- Another possible solution is to apply the ISA separation theorem, which remains valid even for complex variables [167]: 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 [161] the AR-IPA assumption holds (Eq. (181)), 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 $\mathbf{B}_j \in \mathbb{R}^{D\times D_u}$:

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t \qquad\qquad \mathbf{s}_t = \sum_{i=1}^{L_s} \mathbf{F}_i \mathbf{s}_{t-i} + \sum_{j=1}^{L_u} \mathbf{B}_j \mathbf{u}_{t+1-j} + \mathbf{e}_t. \tag{186}$$

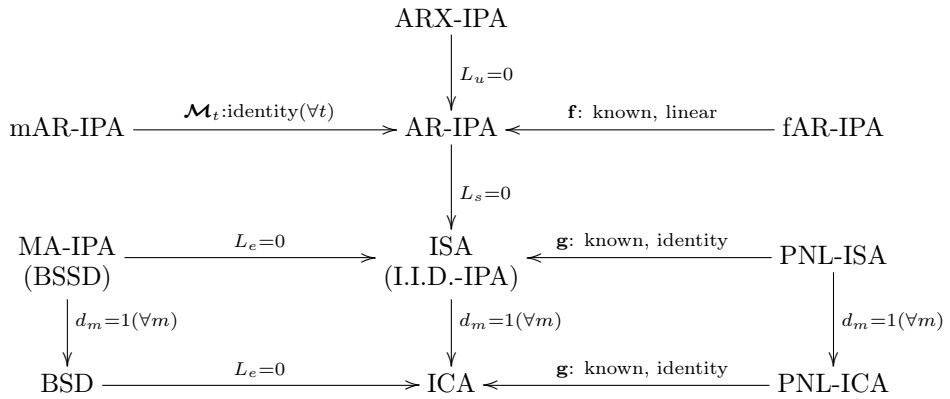


Figure 3: IPA problem family, relations. Arrows point to special cases. For example, 'ISA $\xrightarrow{d_m=1(\forall m)}$ ICA' means that ICA is a special case of ISA, when all the source components are one-dimensional.

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 [161]. Demo: `demo_ARX_IPA.m`.

Partially observed models:

Equations, assumptions: In the *mAR-IPA* (mAR – autoregressive with missing values) problem [157], the AR-IPA assumptions (Eq. (181)) 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' $\boldsymbol{\mathcal{M}}_t : \mathbb{R}^D \mapsto \mathbb{R}^D$ represent the coordinates and the time indices of the non-missing observations:

$$\mathbf{y}_t = \boldsymbol{\mathcal{M}}_t(\mathbf{x}_t), \qquad \mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad \mathbf{s}_t = \sum_{i=1}^{L_s} \mathbf{F}_i \mathbf{s}_{t-i} + \mathbf{e}_t. \tag{187}$$

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 '$\boldsymbol{\mathcal{M}}_t$ = 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 [157]. Demo: `demo_mAR_IPA.m`.

Models with nonparametric dynamics:

Equations, assumptions: In the *fAR-IPA* (fAR – functional autoregressive) problem [165], the *parametric* assumption for the dynamics of the hidden sources is circumvented by functional AR sources:

$$\mathbf{x}_t = \mathbf{A}\mathbf{s}_t, \qquad \mathbf{s}_t = \mathbf{f}(\mathbf{s}_{t-1}, \dots, \mathbf{s}_{t-L_s}) + \mathbf{e}_t. \tag{188}$$

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 [165]. Demo: `demo_fAR_IPA.m`.

Cost function to minimize	Name (`ISA.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'`
$\sum_{m=1}^{M-1} I\left(\mathbf{y}^m,\left[\mathbf{y}^{m+1},...,\mathbf{y}^M\right]\right)$	`'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 17: ISA formulations. $1-4^{th}$ row: equivalent, $5-6^{th}$ row: necessary conditions.

5.2 Estimation via ITE

Having (i) the information theoretical estimators (Section 3), (ii) the ISA/IPA problems and separation principles (Section 5.1) 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 5.2.1 the solution techniques for the ISA task are detailed. Extensions of the ISA problem are in the focus of Section 5.2.2.

5.2.1 ISA

As it has been detailed in Section 5.1.1, the ISA problem can be formulated as the optimization of information theoretical objectives (see Eqs. (173), (174), (175), (176), (177), (178)). In the ITE package,

All the detailed ISA formulations:

- are available by the appropriate choice of the variable `ISA.cost_type` (see Table 17), and
- can be used by *any* entropy/mutual information estimator satisfying the ITE template construction (see Table 2, Table 3, Table 8, Table 9 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 (`ISA.opt_type = 'exhaustive'`), greedy (`ISA.opt_type = 'greedy'`), or the cross-entropy (`ISA.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 17) and the objective is (178), i.e., `ISA.cost_type = 'Ipairwise1d'`. The ITE package supports 4 different spectral clustering methods/implementations (Table 18):
  - the unnormalized cut method (`ISA.opt_type = 'SP1'`), and two normalized cut techniques (`ISA.opt_type = 'SP2'` or `ISA.opt_type = 'SP3'`) [145, 100, 187] – the implementations are purely Matlab/Octave, and
  - a fast, normalized cut implementation [145, 24] in C++ with compilable mex files (`ISA.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 27 (ISA-1)

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

Table 18: 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 sum of entropy based ISA formulation (Eq. (175)); 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`; `ISA.cost_type = 'sumH'`; `ISA.cost_name = 'Renyi_kNN_1tok'`, `ISA.opt_type = 'greedy'`.

Example 28 (ISA-2)

- *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. (176)); estimate the mutual information via the KCCA method; optimize the objective in a greedy way.*
- *Parameters to set:* `unknown_dimensions = 0`; `ISA.cost_type = 'sum-I'`; `ISA.cost_name = 'KCCA'`, `ISA.opt_type = 'greedy'`.

Example 29 (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. (178)); estimate the mutual information using the KGV method; optimize the objective via the NCut normalized cut method.*
- *Parameters to set:* `unknown_dimensions = 1`; `ISA.cost_type = 'Ipairwise1d'`; `ISA.cost_name = 'KGV'`, `ISA.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. (178)), 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 (178), 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 (175), (176) 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 (177) 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 19 (greedy), Table 20 (cross-entropy), Table 21 (exhaustive).

Extending the capabilities of the ITE toolbox: In case of

- 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`.[26]

[26] For example, upon applying objective (175) multiplicative constants are irrelevant (important) in case of equal (different) d_m subspace dimensions.

Cost type (`ISA.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 19: Recommended/chosen optimizers for given subspace dimensions ($\{d_m\}_{m=1}^M$) [`unknown_dimensions=0`] applying greedy [`ISA.opt_type='greedy'`] ISA optimization: `clustering_UD0.m`.

Cost type (`ISA.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 20: Recommended/chosen optimizers for given subspace dimensions ($\{d_m\}_{m=1}^M$) [`unknown_dimensions=0`] applying cross-entropy [`ISA.opt_type='CE'`] ISA optimization: `clustering_UD0.m`.

  - To add a new ISA formulation (`ISA.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 (175), (176)) 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 (`ISA.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 16). To include a new clustering technique, one only has to add it to a new case entry in `estimate_clustering_UD1_S.m`.

5.2.2 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 22. For the obtained subtasks the ITE package provides many efficient estimators (see Table 23):

ICA, complex ICA:

- The fastICA method [63] and its complex variant [13] is one of the most popular ICA approach, it is available in ITE.

Cost type (`ISA.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 21: Recommended/chosen optimizers for given subspace dimensions ($\{d_m\}_{m=1}^M$) [`unknown_dimensions=0`] applying exhaustive [`ISA.opt_type='exhaustive'`] ISA optimization: `clustering_UD0.m`.

- The EASI (equivariant adaptive separation via independence) [19] ICA method family realizes a very exciting online, adaptive approach offering uniform performance w.r.t. the mixing matrix. It is capable of handling the real and the complex case, too.
- As we have seen the search for the demixing matrix in ISA (specifically in ICA) can be restricted to the orthogonal group ($\mathbf{W} \in \mathcal{O}^D$, see Section 5.1.1). Moreover, ortogonal matrices can be written as a product of elementary Jacobi/Givens rotation matrices. The method carries out the search for $\mathbf{W}$ in the ICA problem by the sequential optimization of such elementary rotations on a gradually fined scale. ITE supports Jacobi/Givens based ICA optimization using general entropy and mutual information estimators (`ICA.cost_type = 'sumH'` or `'I'`) for the real case; the pseudo-code of the method is given in Alg. 1.[27] Let us take an example:

  Example 30 (ISA-4)

  - *Goal:*
    - *Task: ISA with known subspace dimensions* $\{d_m\}_{m=1}^M$.
    - *ICA subtask: minimize the mutual information of the estimated coordinate pairs using the KCCA objective; optimize the ICA cost via the Jacobi method,*
    - *ISA subtask (clustering of the ICA elements): apply entopy sum based ISA formulation (Eq. (175)) and estimate the entropy via the Rényi entropy using k-nearest neighbors* ($S = \{1, \ldots, k\}$)*; optimize the ISA objective in a greedy way.*
  - *Parameters to set (see* `demo_ISA.m`*):* `unknown_dimensions = 0`; `ICA.cost_type = 'I'`; `ICA.cost_name = 'KCCA'`; `ICA.opt_type = 'Jacobi1'`; `ISA.cost_type = 'sumH'`; `ISA.cost_name = 'Renyi_kNN_1tok'`; `ISA.opt_type = 'greedy'`.

- An alternative Jacobi optimization method with a different fining scheme in the rotation angle search is also available in ITE, see Alg. 2. The optimization extends the idea of the RADICAL ICA method [80] to general entropy, mutual information objectives. The RADICAL approach can be obtained in ITE by setting `ICA.cost_type = 'sumH'`; `ICA.cost_name = 'Shannon_spacing_V'`; `ICA.opt_type = 'Jacobi2'` (see `demo_ISA.m`).

See `estimate_ICA.m` and `estimate_complex_ICA.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 [71, 118],
- applying [66]
  - 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 [99, 137].

ARX identification: Identification of ARX processes can be carried out by the D-optimal technique of [118] assuming normal-inverted Wishart prior; see `estimate_ARX_IPA.m`.

mAR identification: The

- online Bayesian technique with normal-inverted Wishart prior [71, 118],
- nonlinear least squares [66],
- exact maximum likelihood [66], and
- their combination [66]

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

[27] The optimization extends the idea of the SWICA package [74].

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 22: IPA separation principles.

fAR identification: Identification of fAR processes in ITE can be carried out by the strongly consistent, recursive Nadaraya-Watson estimator [57]; 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 [145, 100, 187] – the implemetations are purely Matlab/Octave, and
- a fast, normalized cut implementation [145, 24] in C++ with compilable mex files.

gaussianization: Gaussianization of the observations can be carried out by the efficient rank method [198], 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 23.

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

Subtask	Estimator	Method
ICA	`estimate_ICA.m`	`'fastICA'`, `'EASI'`, `'Jacobi1'`, `'Jacobi2'`
complex ICA	`estimate_complex_ICA.m`	`'fastICA'`, `'EASI'`
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 23: IPA subtasks and estimators.

Algorithm 1 Jacobi optimization - 1; see `estimate_ICA_Jacobi1.m`.

Input:
  whitened observation $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{d \times T}$,
  ICA cost function on coordinate pairs $J(z_1, z_2) = I(z_1, z_2)$ or $J(z_1, z_2) = H(z_1) + H(z_2)$,
  number of levels $L(= 3 : \text{default value})$, number of sweeps $S(= d)$, number of angles $A(= 90)$.
Notation:
  $\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$. // rotation with angle θ
Initialization:
  estimated demixing matrix $\hat{\mathbf{W}} = \mathbf{I} \in \mathbb{R}^{d \times d}$, estimated source $\hat{\mathbf{E}} = \mathbf{X} \in \mathbb{R}^{d \times T}$.
for $l = 1$ to L **do**
  $a = \left\lceil \frac{A}{2^{L-l}} \right\rceil$. // number of angles at the actual level
  for $s = 1$ to S **do**
    for all $(i_1, i_2) \in \{(i, j) : 1 \leq i < j \leq d\}$ **do**
      $\theta^* = \arg\min_{\theta \in \left\{ \frac{k}{a} \frac{\pi}{2} : k = 0, \ldots, a-1 \right\}} J\left(\mathbf{R}(\theta)\mathbf{e}([i_1, i_2], :)\right)$. // best rotation angle for the $(i_1, i_2)^{th}$ coordinate pair
      Apply the optimal rotation found (θ^*):
        $\hat{\mathbf{W}}([i_1, i_2], :) = \mathbf{R}(\theta^*)\, \hat{\mathbf{W}}([i_1, i_2], :)$,
        $\hat{\mathbf{E}}([i_1, i_2], :) = \mathbf{R}(\theta^*)\, \hat{\mathbf{E}}([i_1, i_2], :)$.
Output: $\hat{\mathbf{W}}$, $\hat{\mathbf{E}}$.

Algorithm 2 Jacobi optimization - 2; see `estimate_ICA_Jacobi2.m`.

Input:
  whitened observation $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{d \times T}$,
  ICA cost function on coordinate pairs $J(z_1, z_2) = I(z_1, z_2)$ or $J(z_1, z_2) = H(z_1) + H(z_2)$,
  number of sweeps $S(= d - 1 : \text{default})$, number of angles $A(= 150)$.
Notation:
  $\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$. // rotation with angle θ
Initialization:
  estimated demixing matrix $\hat{\mathbf{W}} = \mathbf{I} \in \mathbb{R}^{d \times d}$, estimated source $\hat{\mathbf{E}} = \mathbf{X} \in \mathbb{R}^{d \times T}$,
  minimum number of angles $a_{min} = \frac{A}{1.3^{\left\lceil \frac{S}{2} \right\rceil}}$.
for $s = 1$ to S **do**
  if $s > \frac{S}{2}$ **then**
    $a = \left\lfloor a_{min} 1.3^{\left\lceil s - \frac{S}{2} \right\rceil} \right\rfloor$. // number of angles at the actual sweep
  else
    $a = \max\left(30, \lfloor a_{min} \rfloor\right)$. // number of angles at the actual sweep
  for all $(i_1, i_2) \in \{(i, j) : 1 \leq i < j \leq d\}$ **do**
    $\theta^* = \arg\min_{\theta \in \left\{ \frac{k}{a} \frac{\pi}{2} : k = 0, \ldots, a-1 \right\}} J\left(\mathbf{R}(\theta)\mathbf{e}([i_1, i_2], :)\right)$. // best rotation angle for the $(i_1, i_2)^{th}$ coordinate pair
    Apply the optimal rotation found (θ^*):
      $\hat{\mathbf{W}}([i_1, i_2], :) = \mathbf{R}(\theta^*)\, \hat{\mathbf{W}}([i_1, i_2], :)$,
      $\hat{\mathbf{E}}([i_1, i_2], :) = \mathbf{R}(\theta^*)\, \hat{\mathbf{E}}([i_1, i_2], :)$.
Output: $\hat{\mathbf{W}}$, $\hat{\mathbf{E}}$.

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

[a] See Table 1.

co.MSTmethod	Method	Environment
'pmtk3_Prim'	Prim algorithm (pmtk3)	Matlab, Octave
'pmtk3_Kruskal'	Kruskal algorithm (pmtk3)	Matlab, Octave

Table 25: Minimum spanning tree (MST) methods. The main MST function is `compute_MST.m`.

- 4 different alternatives for *k-nearest neighbor* estimation (Table 24):
  - 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 kNN method applied for the estimation can be chosen by setting `co.method` to `'knnFP1'`, `'knnFP2'`, `'knnsearch'`, or `'ANN'`. For examples, see the initialization files of
  - entropy: `'Shannon_kNN_k'`, `'Renyi_kNN_1tok'`, `'Renyi_kNN_k'`, `'Renyi_kNN_S'`, `'Renyi_weightedkNN'`, `'Tsallis_kNN_k'`, `'SharmaM_kNN_k'`,
  - divergence: `'L2_kNN_k'`, `'Renyi_kNN_k'`, `'Tsallis_kNN_k'`, `'KL_kNN_kiTi'`, `'Hellinger_kNN_k'`, `'KL_kNN_k'`, `'Bhattacharyya_kNN_k'`, `'symBregman_kNN_k'`, `'Bregman_kNN_k'`, `'ChiSquare_kNN_k'`, `'SharmaM_kNN_k'`,
  - cross-quantity: `'CE_kNN_k'`,
  - kernel on distributions: `'Bhattacharyya_kNN_k'`, `'PP_kNN_k'`.

  The central function of kNN computations is `kNN_squared_distances.m`.
- 2 techniques for *minimum spanning tree* computation (Table 25): 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 measure and cross quantity estimation), it sufficient to the add the new method to `kNN_squared_distances.m` or `compute_MST.m`.

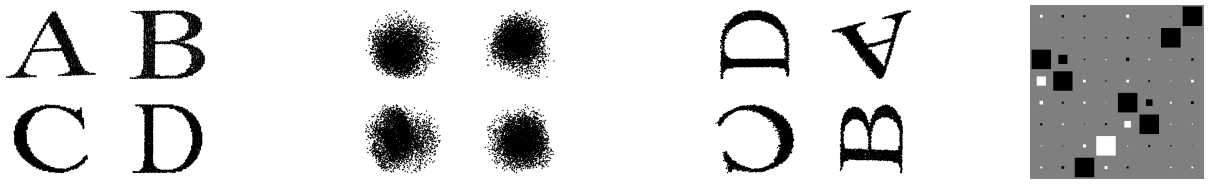
5.3 Performance Measure, the Amari-index

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 [179]. 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

$$\mathbf{G} = \hat{\mathbf{W}}_{\mathrm{ISA}}\mathbf{A} \tag{189}$$

is a block-permutation matrix (also called block-scaling matrix [178]). This property can also be measured for source components with different dimensions by a simple extension [165] of the Amari-index [4], that we present below. Namely,



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

Figure 4: 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 2×2 blocks.

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 ($\mathbf{G} = \left[\mathbf{G}^{ij}\right]_{i,j=1,\ldots,M}$) and define g^{ij} as the ℓ_q norm[28] of the elements of the matrix $\mathbf{G}^{ij} \in \mathbb{R}^{d_i \times d_j}$, weighted with V_{ij}:

$$g^{ij} = V_{ij} \left(\sum_{k=1}^{d_i} \sum_{l=1}^{d_j} |\left(\mathbf{G}^{ij}\right)_{k,l}|^q \right)^{\frac{1}{q}}. \tag{190}$$

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

$$r_{\mathbf{V},q}(\mathbf{G}) := \frac{1}{2M(M-1)} \left[\sum_{i=1}^{M} \left(\frac{\sum_{j=1}^{M} g^{ij}}{\max_j g^{ij}} - 1 \right) + \sum_{j=1}^{M} \left(\frac{\sum_{i=1}^{M} g^{ij}}{\max_i g^{ij}} - 1 \right) \right]. \tag{191}$$

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 (191) 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. (191)] 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. 4 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 5.1.2. The demo files in the ITE toolbox (see Table 22) contain detailed examples for the usage of the Amari-index in the extensions of ISA.

5.4 Dataset-, Model Generators

One can generate observations from the ISA model and its extensions (Section 5.1.2) by the functions listed in Table 26. The sources/driving noises can be chosen from many different types in ITE (see `sample_subspaces.m`):

3D-geom: In the *3D-geom* test [117] $\mathbf{e}^m$s are random variables uniformly distributed on 3-dimensional geometric forms ($d_m = 3$, $M \leq 6$), see Fig. 5(a). The dataset generator is `sample_subspaces_3D_geom.m`.

Aw, ABC, GreekABC: In the $A\omega$ database [160] 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* ($M \leq 26$) [116] and the *GreekABC* ($M \leq 24$) [160] subsets. For illustration, see Fig. 5(d). The dataset generators are called `sample_subspaces_Aw.m`, `sample_subspaces_ABC.m` and `sample_subspaces_GreekABC.m`, respectively.

mosaic: The *mosaic* test [169] 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

[28] Alternative norms could also be used.

other words, 2-dimensional images are considered as density functions. For illustration, see Fig. 5(h). The dataset generator is `sample_subspaces_mosaic.m`.

IFS: Here [171], components $\mathbf{s}^m$ are realizations of IFS[29] based 2-dimensional ($d = 2$) self-similar structures. For all m a $(\{\mathbf{h}_k\}_{k=1,\dots,K}, \mathbf{p} = (p_1, \dots, p_K), \mathbf{v}_1\}$ triple is chosen, where

- $\mathbf{h}_k : \mathbb{R}^2 \to \mathbb{R}^2$ 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 $\{1, \dots, K\}$ ($\sum_{k=1}^K p_k = 1, p_k \geq 0$), 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 $k(t) \in \{1, \dots, K\}$ 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 $\{\mathbf{v}_1, \dots, \mathbf{v}_T\}$ was taken as a hidden source component $\mathbf{s}^m$ and this way 9 components ($M = 9$, $D = 18$) were constructed (see Fig. 5(c)). The generator of the dataset is `sample_subspaces_IFS.m`.

ikeda: In the *ikeda* test [165], the hidden $\mathbf{s}_t^m = [s_{t,1}^m, s_{t,2}^m] \in \mathbb{R}^2$ sources realize the ikeda map

$$s_{t+1,1}^m = 1 + \lambda_m[s_{t,1}^m \cos(w_t^m) - s_{t,2}^m \sin(w_t^m)], \tag{192}$$

$$s_{t+1,2}^m = \lambda_m[s_{t,1}^m \sin(w_t^m) + s_{t,2}^m \cos(w_t^m)], \tag{193}$$

where λ_m is a parameter of the dynamical system and

$$w_t^m = 0.4 - \frac{6}{1 + (s_{t,1}^m)^2 + (s_{t,2}^m)^2}. \tag{194}$$

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. 5(f) for illustration. Observation can be generated from this dataset using `sample_subspaces_ikeda.m`.

lorenz: In the *lorenz* dataset [169], the sources ($\mathbf{s}^m$) correspond to 3-dimensional ($d_m = 3$) deterministic chaotic time series, the so-called Lorenz attractor [85] 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:

$$\dot{x}_t = a(y_t - x_t), \tag{195}$$

$$\dot{y}_t = x_t(b - z_t) - y_t, \tag{196}$$

$$\dot{z}_t = x_t y_t - c z_t. \tag{197}$$

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. 5(g).

all-k-independent: In the *all-k-independent* database [116, 166], the d_m-dimensional hidden components $\mathbf{v} := \mathbf{e}^m$ are created as follows: coordinates v_i ($i = 1, \dots, k$) are independent uniform random variables on the set {0,...,k-1}, whereas v_{k+1} is set to $mod(v_1 + \dots + v_k, k)$. In this construction, every k-element subset of $\{v_1, \dots, v_{k+1}\}$ 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 $\mathbf{0} \to \mathbf{e}_1 \to \mathbf{e}_1 + \mathbf{e}_2 \to \dots \to \mathbf{e}_1 + \dots + \mathbf{e}_{d_m}$ (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 ($d_1 = \dots = d_M$); in case of the *multiD$_1$-...-D$_M$-geom* dataset, the d_m subspace dimensions can be different. For illustration, see Fig. 5(e). The associated dataset generator is called `sample_subspaces_multiD_geom.m`.

[29] IFS stands for iterated function system.

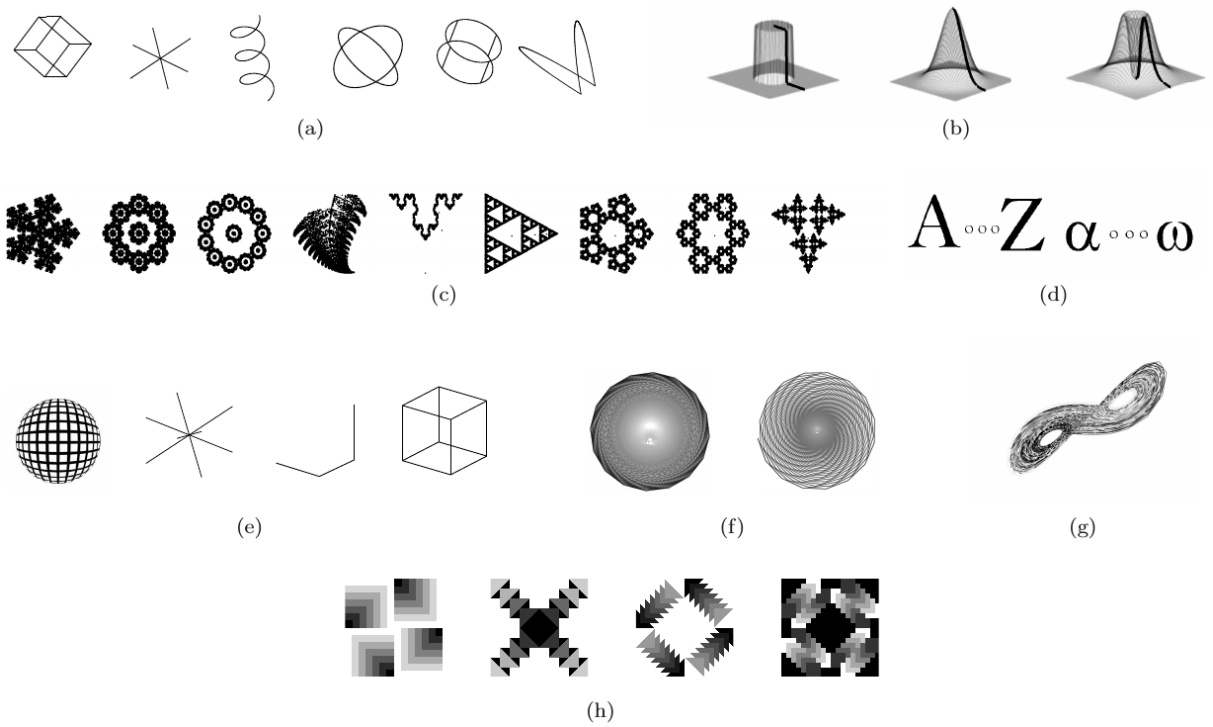


Figure 5: Illustration of the *3D-geom* (a), *multiD-spherical (multiD$_1$-...-D$_M$-spherical)* (b), *IFS* (c), *Aw* (subset on the left: *ABC*, right: *GreekABC*) (d), *multiD-geom (multiD$_1$-...-D$_M$-geom)* (e), *ikeda* (f), *lorenz* (g), and *mosaic* (h) datasets.

multiD-spherical (multiD$_1$-...-D$_M$-spherical): In this case hidden sources $\mathbf{e}^m$ are spherical random variables [37]. 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 ρ is a non-negative scalar random variable independent of $\mathbf{u}$, they can be given by means of ρ. 3 pieces of stochastic representations ρ were chosen: ρ was uniform on $[0, 1]$, exponential with parameter $\mu = 1$ and lognormal with parameters $\mu = 0$, $\sigma = 1$. For illustration, see Fig. 5(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 27 and Table 28. The `plot_subspaces.m` function can be used to plot the databases (samples/estimations).

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 26: 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 27: 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 28: Generators of the datasets. The high-level sampling function of the datasets is `sample_subspaces.m`.

6 Quick Tests for the Estimators

Beyond IPA (see Section 5), ITE provides quick tests to study the efficiency of the different information theoretical estimators. The tests can be grouped into four categories, described in the following sections.

6.1 Analytical Expression versus Estimaton

ITE provides tests for analytical expressions versus their estimated values as a function of the sample number, see Table 29. The analytical formulas are based on the

$$H(\mathbf{Ay}) = H(\mathbf{y}) + \log\left(|\mathbf{A}|\right), \qquad H_{\mathrm{R},\alpha}(\mathbf{Ay}) = H_{\mathrm{R},\alpha}(\mathbf{y}) + \log\left(|\mathbf{A}|\right), \tag{198}$$

entropy transformation rules and the following relations [25, 149, 44, 189, 109, 65, 94, 103] (derivation of formula (199), (203), (204), (215), (217), (228), (229), (230) and (231) can be found in Section F)

Entropy:

$$\mathbb{R}^d \ni \mathbf{y} \sim U[\mathbf{a},\mathbf{b}], \qquad H(\mathbf{y}) = \log\left[\prod_{i=1}^{d}(b_i - a_i)\right], \tag{199}$$

$$\mathbb{R}^d \ni \mathbf{y} \sim N(\mathbf{m},\mathbf{\Sigma}), \qquad H(\mathbf{y}) = \frac{1}{2}\log\left[(2\pi e)^d|\mathbf{\Sigma}|\right] = \frac{1}{2}\log\left(|\mathbf{\Sigma}|\right) + \frac{d}{2}\log(2\pi) + \frac{d}{2}, \tag{200}$$

$$\mathbb{R}^d \ni \mathbf{y} \sim U[\mathbf{a},\mathbf{b}], \qquad H_{\mathrm{R},\alpha}(\mathbf{y}) = \log\left[\prod_{i=1}^{d}(b_i - a_i)\right], \tag{201}$$

$$\mathbb{R}^d \ni \mathbf{y} \sim N(\mathbf{m},\mathbf{\Sigma}), \qquad H_{\mathrm{R},\alpha}(\mathbf{y}) = \log\left[(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}\right] - \frac{d\log(\alpha)}{2(1-\alpha)}, \tag{202}$$

$$\mathbb{R}^d \ni \mathbf{y} \sim U[\mathbf{a},\mathbf{b}], \qquad H_{\mathrm{T},\alpha}(\mathbf{y}) = \frac{\left[\prod_{i=1}^{d}(b_i - a_i)\right]^{1-\alpha} - 1}{1-\alpha}, \tag{203}$$

$$\mathbb{R}^d \ni \mathbf{y} \sim N(\mathbf{m},\mathbf{\Sigma}), \qquad H_{\mathrm{T},\alpha}(\mathbf{y}) = \frac{\frac{\left[(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}\right]^{1-\alpha}}{\alpha^{\frac{d}{2}}} - 1}{1-\alpha}, \tag{204}$$

$$\mathbb{R} \ni y \sim U[a,b], \qquad H_{\Phi(u)=u^c, w(u)=\mathbb{I}_{[a,b]}(u)}(y) = \frac{1}{(b-a)^c} \quad (c \geq 1), \tag{205}$$

Mutual information:

$$\mathbb{R}^d \ni \mathbf{y} \sim N(\mathbf{m},\mathbf{\Sigma}) \qquad I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = \frac{1}{2}\log\left(\frac{\prod_{m=1}^{M}\det(\mathbf{\Sigma}_m)}{\det(\mathbf{\Sigma})}\right), \tag{206}$$

$$\mathbb{R}^d \ni \mathbf{y} \sim N(\mathbf{m},\mathbf{\Sigma}), \qquad t_1 = -\frac{\alpha\log(|\mathbf{\Sigma}|)}{2}, \quad t_2 = -\frac{(1-\alpha)\log\left(\prod_{i=1}^{d}\Sigma_{ii}\right)}{2}, \tag{207}$$

$$t_3 = -\frac{\log\left(\det\left[\alpha\mathbf{\Sigma}^{-1} + (1-\alpha)diag\left(\frac{1}{\Sigma_{11}},\ldots,\frac{1}{\Sigma_{dd}}\right)\right]\right)}{2}, \tag{208}$$

$$I_{\mathrm{R},\alpha}\left(y^1,\ldots,y^d\right) = \frac{t_1+t_2+t_3}{\alpha-1}, \tag{209}$$

Divergence:

$$\mathbb{R}^d \ni f_m = N(\mathbf{m}_m,\mathbf{\Sigma}_m), \qquad D(f_1,f_2) = \frac{1}{2}\left[\log\left(\frac{|\mathbf{\Sigma}_2|}{|\mathbf{\Sigma}_1|}\right) + tr\left(\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1\right)\right. \tag{210}$$

$$\left. + (\mathbf{m}_1-\mathbf{m}_2)^*\mathbf{\Sigma}_2^{-1}(\mathbf{m}_1-\mathbf{m}_2) - d\right], \tag{211}$$

$$\mathbb{R}^d \ni f_m = N(\mathbf{m}_m, \mathbf{\Sigma}_m), \qquad \mathbf{m} = \mathbf{m}_1 - \mathbf{m}_2, \quad \mathbf{M}_\alpha = \alpha\mathbf{\Sigma}_2 + (1-\alpha)\mathbf{\Sigma}_1, \tag{212}$$

$$D_{\mathrm{R},\alpha}(f_1, f_2) = \frac{\alpha}{2}\mathbf{m}^*\mathbf{M}_\alpha^{-1}\mathbf{m} - \frac{1}{2(\alpha-1)}\log\left(\frac{|\mathbf{M}_\alpha|}{|\mathbf{\Sigma}_1|^{1-\alpha}|\mathbf{\Sigma}_2|^\alpha}\right), \tag{213}$$

$$D_{\mathrm{T},\alpha}(f_1, f_2) = \frac{e^{(\alpha-1)D_{\mathrm{R},\alpha}(f_1,f_2)-1}}{\alpha-1}, \tag{214}$$

$$\mathbb{R}^d \ni f_m = U[\mathbf{0}, \mathbf{a}_m](\mathbf{a}_2 \le \mathbf{a}_1), \qquad D_{\mathrm{L}}(f_1, f_2) = \sqrt{\frac{1}{\prod_{i=1}^d (\mathbf{a}_2)_i} - \frac{1}{\prod_{i=1}^d (\mathbf{a}_1)_i}}, \tag{215}$$

$$\mathbb{R}^d \ni \mathbf{y}^m \sim f_m = N\left(\mathbf{m}_m, \sigma_m^2\mathbf{I}\right), \quad D^\pi_{\mathrm{JR},2}(f_1, \ldots, f_M) = a\left(\pi, \{\mathbf{m}_m, \sigma_m^2\mathbf{I}\}_{m=1}^M\right) - \sum_{m=1}^M \pi_m a\left(\pi_m, \mathbf{m}_m, \sigma_m^2\mathbf{I}\right), (M=2), \tag{216}$$

$$\mathbb{R}^d \ni f_m = U[\mathbf{0}, \mathbf{a}_m](\mathbf{a}_1 \le \mathbf{a}_2), \qquad D_{\mathrm{NB},\alpha}(f_1, f_2) = \frac{1}{\alpha-1}\left(\left[\prod_{i=1}^d (\mathbf{a}_1)_i\right]^{1-\alpha} - \left[\prod_{i=1}^d (\mathbf{a}_2)_i\right]^{1-\alpha}\right), \tag{217}$$

$$\mathbb{R}^d \ni f_m = U[\mathbf{0}, \mathbf{a}_m](\mathbf{a}_1 \le \mathbf{a}_2), \qquad D_{\chi^2}(f_1, f_2) = \frac{\prod_{i=1}^d (\mathbf{a}_2)_i}{\prod_{i=1}^d (\mathbf{a}_1)_i} - 1, \tag{218}$$

$$\mathbb{R}^d \ni f_m = N(\mathbf{m}_m, \mathbf{I}) \qquad D_{\chi^2}(f_1, f_2) = e^{(\mathbf{m}_2-\mathbf{m}_1)^*(\mathbf{m}_2-\mathbf{m}_1)} - 1, \tag{219}$$

$$\mathbb{R}^d \ni f_m = N(\mathbf{m}_m, \mathbf{\Sigma}_m), \qquad D_H(f_1, f_2) = \sqrt{1 - \frac{|\mathbf{\Sigma}_1|^{\frac{1}{4}}|\mathbf{\Sigma}_2|^{\frac{1}{4}}}{\left|\frac{\mathbf{\Sigma}_1+\mathbf{\Sigma}_2}{2}\right|^{\frac{1}{2}}} e^{-\frac{(\mathbf{m}_1-\mathbf{m}_2)^*\left(\frac{\mathbf{\Sigma}_1+\mathbf{\Sigma}_2}{2}\right)^{-1}(\mathbf{m}_1-\mathbf{m}_2)}{8}}}, \tag{220}$$

Cross quantity:

$$\mathbb{R}^d \ni f_m = N(\mathbf{m}_m, \mathbf{\Sigma}_m), \quad C_{\mathrm{CE}}(f_1, f_2) = \frac{1}{2}\left[d\log(2\pi) + \log(|\mathbf{\Sigma}_2|) + tr\left(\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1\right) + \right. \tag{221}$$

$$\left. (\mathbf{m}_1 - \mathbf{m}_2)^*\mathbf{\Sigma}_2^{-1}(\mathbf{m}_1 - \mathbf{m}_2)\right], \tag{222}$$

Kernel on distribution:

$$\mathbb{R}^d \ni f_m = N\left(\mathbf{m}_m, \mathbf{\Sigma}_m\right), \qquad K_{\exp}\left(f_1, f_2\right) = \frac{e^{-\frac{1}{2}(\mathbf{m}_1-\mathbf{m}_2)^*\left(\mathbf{\Sigma}_1+\mathbf{\Sigma}_2+\gamma^{-1}\mathbf{I}\right)^{-1}(\mathbf{m}_1-\mathbf{m}_2)}}{|\gamma\mathbf{\Sigma}_1 + \gamma\mathbf{\Sigma}_2 + \mathbf{I}|^{\frac{1}{2}}}, \tag{223}$$

$$\mathbb{R}^d \ni \mathbf{y}^m \sim f_m = N\left(\mathbf{m}_m, \mathbf{\Sigma}_m\right), \qquad \mathbf{m}' = \mathbf{\Sigma}_1^{-1}\mathbf{m}_1 + \mathbf{\Sigma}_2^{-1}\mathbf{m}_2, \tag{224}$$

$$\mathbf{\Sigma}' = \left(\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}\right)^{-1}, \tag{225}$$

$$K_{\mathrm{PP},\rho}\left(f_1, f_2\right) = (2\pi)^{\frac{(1-2\rho)d}{2}}\rho^{-\frac{d}{2}}\left|\mathbf{\Sigma}'\right|^{\frac{1}{2}}\left|\mathbf{\Sigma}_1\right|^{-\frac{\rho}{2}}\left|\mathbf{\Sigma}_2\right|^{-\frac{\rho}{2}} \tag{226}$$

$$\times\, e^{-\frac{\rho}{2}\left[\mathbf{m}_1^*\mathbf{\Sigma}_1^{-1}\mathbf{m}_1 + \mathbf{m}_2^*\mathbf{\Sigma}_2^{-1}\mathbf{m}_2 - \left(\mathbf{m}'\right)^*\left(\mathbf{\Sigma}'\right)\left(\mathbf{m}'\right)\right]}, \tag{227}$$

$$\mathbb{R}^d \ni f_m = N\left(\mathbf{m}_m, \sigma_m^2\mathbf{I}\right), \qquad K_{\mathrm{EJR1},u,2}(f_1, f_2) = e^{-u\left[a\left(\left[\frac{1}{2},\frac{1}{2}\right], \{\mathbf{m}_m, \sigma_m^2\mathbf{I}\}_{m=1}^2\right)\right]}, \tag{228}$$

$$\mathbb{R}^d \ni f_m = N\left(\mathbf{m}_m, \sigma_m^2\mathbf{I}\right), \qquad K_{\mathrm{EJR2},u,2}(f_1, f_2) = e^{-u\left[a\left(\left[\frac{1}{2},\frac{1}{2}\right], \{\mathbf{m}_m, \sigma_m^2\mathbf{I}\}_{m=1}^2\right) - \frac{1}{2}\sum_{m=1}^2 a\left(\frac{1}{2}, \mathbf{m}_m, \sigma_m^2\mathbf{I}\right)\right]}, \tag{229}$$

$$\mathbb{R}^d \ni f_m = N\left(\mathbf{m}_m, \sigma_m^2\mathbf{I}\right), \qquad K_{\mathrm{EJT1},u,2}(f_1, f_2) = e^{-u\left[1 - e^{-a\left(\left[\frac{1}{2},\frac{1}{2}\right], \{\mathbf{m}_m, \sigma_m^2\mathbf{I}\}_{m=1}^2\right)}\right]}, \tag{230}$$

$$\mathbb{R}^d \ni f_m = N\left(\mathbf{m}_m, \sigma_m^2\mathbf{I}\right), \qquad K_{\mathrm{EJT2},u,2}(f_1, f_2) = e^{-u\left(1 - e^{-a\left(\left[\frac{1}{2},\frac{1}{2}\right], \{\mathbf{m}_m, \sigma_m^2\mathbf{I}\}_{m=1}^2\right)} - \frac{1}{2}\sum_{m=1}^2 \left[1 - e^{-a\left(\frac{1}{2}, \mathbf{m}_m, \sigma_m^2\mathbf{I}\right)}\right]\right)}, \tag{231}$$

where $U[\mathbf{a}, \mathbf{b}]$ is the uniform distribution on the rectangle $[\mathbf{a}, \mathbf{b}] = \times_{i=1}^d [a_i, b_i]$, $N(\mathbf{m}, \mathbf{\Sigma})$ denotes the normal distribution with mean $\mathbf{m}$, covariance matrix $\mathbf{\Sigma}$ and

$$a\left(\mathbf{w}, \{\mathbf{m}_m, \sigma_m^2\mathbf{I}\}_{m=1}^M\right) = H_2\left(\sum_{m=1}^M w_m N\left(\mathbf{m}_m, \sigma_m^2\mathbf{I}\right)\right) = -\log\left[\sum_{m_1,m_2=1}^M w_{m_1}w_{m_2}n\left(\mathbf{m}_{m_1} - \mathbf{m}_{m_2}, \left(\sigma_{m_1}^2 + \sigma_{m_2}^2\right)\mathbf{I}\right)\right], \tag{232}$$

$$n(\mathbf{m},\mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\mathbf{\Sigma}|}} e^{-\frac{1}{2}\mathbf{m}^*\mathbf{\Sigma}^{-1}\mathbf{m}}, \tag{233}$$

$\sum_{m=1}^{M} w_m = 1$, $w_m \geq 0$; for matrices $|\cdot|$ denotes the absolute value of the determinant, $tr(\cdot)$ is the trace, Σ_{ii} denotes the $(i,i)^{th}$ element of matrix $\mathbf{\Sigma}$, the $\mathbf{a} \leq \mathbf{b}$ relation is meant coordinate-wise, n is the Gaussian density at zero.

Exponential Family. There exist exciting analytical expressions for information theoretical quantities in the exponential family. An exponential family [16] is a set of density functions (absolutely continuous to measure ν[30]) expressed canonically as

$$f_F(\mathbf{u};\boldsymbol{\theta}) = e^{\langle t(\mathbf{u}),\boldsymbol{\theta}\rangle - F(\boldsymbol{\theta}) + k(\mathbf{u})}, \tag{234}$$

where

- function $F(\boldsymbol{\theta}) = -\log\left[\int e^{\langle t(\mathbf{u}),\boldsymbol{\theta}\rangle + k(\mathbf{u})}\mathrm{d}\mathbf{u}\right]$, the so-called log-normalizer (or partition function, or cumulant function) that characterizes the family,
- $t(\mathbf{u})$ denotes sufficient statistics,
- $\mathbf{\Theta}$ is the natural parameter space ($\boldsymbol{\theta} \in \mathbf{\Theta}$),
- $k(\mathbf{u})$ is an auxiliary function defining the carrier measure ξ [$\mathrm{d}\xi(\mathbf{u}) = e^{k(\mathbf{u})}\mathrm{d}\nu(\mathbf{u})$].

Exponential families include many important examples such as the normal, gamma, beta, exponential, Wishart, Weibull distributions. For example, in the

- normal case ($N(\mathbf{m},\mathbf{\Sigma})$) one gets

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = \left(\mathbf{\Sigma}^{-1}\mathbf{m}, \frac{1}{2}\mathbf{\Sigma}^{-1}\right), \qquad F(\boldsymbol{\theta}) = \frac{1}{4}tr\left(\boldsymbol{\theta}_2^{-1}\boldsymbol{\theta}_1\boldsymbol{\theta}_1^*\right) - \frac{1}{2}\log\det(\boldsymbol{\theta}_2) + \frac{d}{2}\log(\pi), \tag{235}$$

$$t(\mathbf{u}) = (\mathbf{u}, -\mathbf{u}\mathbf{u}^*), \qquad k(\mathbf{u}) = 0. \tag{236}$$

- isotropic normal case ($N(\mathbf{m},\mathbf{I})$, $\mathbf{m} \in \mathbb{R}^d$):

$$\boldsymbol{\theta} = \mathbf{m}, \qquad F(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_2^2, \tag{237}$$

$$t(\mathbf{u}) = \mathbf{u}, \qquad k(\mathbf{u}) = \frac{d}{2}\log(2\pi) - \frac{1}{2}\|\mathbf{u}\|_2^2. \tag{238}$$

Analytical expressions in the exponential family (see Table 30):

- **Entropy**: As it has been recently shown [103], for the exponential family quantity I_α [see Eq. (277)] can be analytically expressed as

$$I_\alpha = e^{F(\alpha\boldsymbol{\theta})-\alpha F(\boldsymbol{\theta})}\int f(\mathbf{u};\alpha\boldsymbol{\theta})e^{(\alpha-1)k(\mathbf{u})}\mathrm{d}\mathbf{u} \overset{\text{if } k(\mathbf{u})=0(\forall\mathbf{u})}{=} e^{F(\alpha\boldsymbol{\theta})-\alpha F(\boldsymbol{\theta})}. \tag{239}$$

  Here, we require that (i) $k \equiv 0$ and (ii) $\alpha\boldsymbol{\theta} \in \mathbf{\Theta}$; the latter holds, for example, if $\mathbf{\Theta}$ is a convex *cone*. Having obtained a formula for I_α, the Sharma-Mittal entropy can be computed as

$$H_{\mathrm{SM},\alpha,\beta}(\mathbf{y}) = \frac{1}{1-\beta}\left[I_\alpha^{\frac{1-\beta}{1-\alpha}} - 1\right]. \tag{240}$$

  The entropy estimator is called '`SharmaM_expF`' in ITE. Specially,

$$\mathbb{R}^d \ni \mathbf{y} \sim N(\mathbf{m},\mathbf{\Sigma}), \qquad H_{\mathrm{SM},\alpha,\beta}(\mathbf{y}) = \frac{1}{1-\beta}\left(\frac{\left[(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}\right]^{1-\beta}}{\alpha^{\frac{d(1-\beta)}{2(1-\alpha)}}} - 1\right). \tag{241}$$

[30] The standard choice for ν is the Lebesgue- or the counting measure when we obtain continuous and discrete distributions, respectively.)

As the limit of the Sharma-Mittal entropy one can obtain [102, 103] many important special cases ($k \equiv 0$ below):

$$H_{\mathrm{R},\alpha}(\mathbf{y}) = \frac{1}{1-\alpha}\left[F(\alpha\boldsymbol{\theta}) - \alpha F(\boldsymbol{\theta})\right], \tag{242}$$

$$H_{\mathrm{T},\alpha}(\mathbf{y}) = \frac{1}{1-\alpha}\left[e^{F(\alpha\boldsymbol{\theta})-\alpha F(\boldsymbol{\theta})} - 1\right], \tag{243}$$

$$H(\mathbf{y}) = F(\boldsymbol{\theta}) - \langle \boldsymbol{\theta}, \nabla F(\boldsymbol{\theta})\rangle, \tag{244}$$

where $\mathbf{y} \sim f_F(\cdot\,;\boldsymbol{\theta})$. The corresponding estimators are called '`Renyi_expF`', '`Tsallis_expF`' and '`Shannon_expF`' in ITE.

- **Cross-entropy**: As it has been shown [102], if $f_1 = f_F(\cdot\,;\boldsymbol{\theta}_1)$, $f_2 = f_F(\cdot\,;\boldsymbol{\theta}_2)$ and $k \equiv 0$, then

$$C_{\mathrm{CE}}(f_1, f_2) = F(\boldsymbol{\theta}_2) - \langle \boldsymbol{\theta}_2, \nabla F(\boldsymbol{\theta}_1)\rangle. \tag{245}$$

  The estimator based on this relation is called '`CE_expF`' in ITE.

  Note: specifically, if $f_1 = f_2$, i.e., $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$, then Eq. (245) reduces to Eq. (244).

- **Divergence**: One can also obtain analytical expression in the exponential family for the
  - Sharma-Mittal divergence. Namely, if $f_1 = f_F(\cdot\,;\boldsymbol{\theta}_1)$ and $f_2 = f_F(\cdot\,;\boldsymbol{\theta}_2)$, then it can be shown [103] that the α-divergence ingredient (D_{temp1} below, see also (70)) is

$$D_{\mathrm{temp1}}(\alpha) = e^{-J_{\mathrm{F},\alpha}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)}, \tag{246}$$

$$J_{\mathrm{F},\alpha}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = \alpha F(\boldsymbol{\theta}_1) + (1-\alpha)F(\boldsymbol{\theta}_2) - F(\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2). \tag{247}$$

    Here we require that $\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2 \in \boldsymbol{\Theta}$. Since $\boldsymbol{\Theta}$ is convex, this is guaranteed, for example, if $\alpha \in (0,1)$. Specially, if $f_m = N(\mathbf{m}_m, \boldsymbol{\Sigma}_m)$ $(m = 1, 2)$, then

$$\boldsymbol{\Sigma}_\alpha = \left[\alpha\boldsymbol{\Sigma}_1^{-1} + (1-\alpha)\boldsymbol{\Sigma}_2^{-1}\right]^{-1}, \tag{248}$$

$$\mathbf{m} = \mathbf{m}_2 - \mathbf{m}_1, \tag{249}$$

$$J_{\mathrm{F},\alpha}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) = \frac{1}{2}\left(\log\left[\frac{|\boldsymbol{\Sigma}_1|^\alpha\,|\boldsymbol{\Sigma}_2|^{1-\alpha}}{|\boldsymbol{\Sigma}_\alpha|}\right] + \alpha(1-\alpha)\mathbf{m}^*\boldsymbol{\Sigma}_\alpha^{-1}\mathbf{m}\right). \tag{250}$$

    The estimator is called '`SharmaM_expF`' in ITE.
  - Kullback-Leibler divergence: If $f_1 = f_F(\cdot\,;\boldsymbol{\theta}_1)$, $f_2 = f_F(\cdot\,;\boldsymbol{\theta}_2)$, then their Kullback-Leibler divergence is equal to the Bregman distance (B_F) defined by the log-normalizer (F) on the swapped natural parameters ($\boldsymbol{\theta}_2$, $\boldsymbol{\theta}_1$) [101]:

$$D(f_1, f_2) = B_F(\boldsymbol{\theta}_2, \boldsymbol{\theta}_1), \tag{251}$$

    where

$$B_F(\mathbf{p}, \mathbf{q}) = F(\mathbf{p}) - F(\mathbf{q}) - \langle \mathbf{p} - \mathbf{q}, \nabla F(\mathbf{q})\rangle. \tag{252}$$

    The name of the corresponding estimator is '`KL_expF`' in ITE.
  - Pearson χ^2 divergence: It has been shown recently [104] that if $f_1 = f_F(\cdot\,;\boldsymbol{\theta}_1)$ and $f_2 = f_F(\cdot\,;\boldsymbol{\theta}_2)$, then

$$D_{\chi^2}(f_1, f_2) = e^{F(2\boldsymbol{\theta}_1-\boldsymbol{\theta}_2)-[2F(\boldsymbol{\theta}_1)-F(\boldsymbol{\theta}_2)]} - 1, \tag{253}$$

    provided that $2\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2 \in \boldsymbol{\Theta}$, which holds, for example, if $\boldsymbol{\Theta}$ is affine. The estimator is called '`ChiSquare_expF`' in ITE. Expression (253) follows from a more general relation [104]: let us suppose that $f_1 = f_F(\cdot\,;\boldsymbol{\theta}_1)$ and $f_2 = f_F(\cdot\,;\boldsymbol{\theta}_2)$, then

$$D_{\mathrm{temp4}}(a, b) = \int_{\mathbb{R}^d} [f_1(\mathbf{u})]^a\,[f_2(\mathbf{u})]^b\,\mathrm{d}\mathbf{u} = e^{F(a\boldsymbol{\theta}_1+b\boldsymbol{\theta}_2)-[aF(\boldsymbol{\theta}_1)+bF(\boldsymbol{\theta}_2)]}, \tag{254}$$

    provided that $a + b = 1$ and $a\boldsymbol{\theta}_1 + b\boldsymbol{\theta}_2 \in \boldsymbol{\Theta}$. The latter condition holds, e.g., if $\boldsymbol{\Theta}$ is affine. D_{temp4} is a reparameterization of D_{temp2}, see Eq. (73).

ITE supports the easy embedding of new exponential families: it is sufficient to add the MLE ($\hat{\boldsymbol{\theta}}$), log-normalizer ($F$), or its gradient ($\nabla F$) to `expF_MLE.m`, `expF_F.m` and `expF_gradF`, respectively.

Estimated quantity	Formula	Quick test
Shannon entropy (H)	(199), (200)	`quick_test_HShannon.m`
Rényi entropy ($H_{\mathrm{R},\alpha}$)	(201), (202)	`quick_test_HRenyi.m`
Tsallis entropy ($H_{\mathrm{T},\alpha}$)	(203), (204)	`quick_test_HTsallis.m`
Sharma-Mittal entropy ($H_{\mathrm{SM},\alpha,\beta}$)	(241)	`quick_test_HSharmaM.m`
Φ-entropy ($H_{\Phi,w}$)	(205)	`quick_test_HPhi.m`
mutual information (I)	(206)	`quick_test_IShannon.m`
Rényi mutual information ($I_{\mathrm{R},\alpha}$)	(209)	`quick_test_IRenyi.m`
Kullback-Leibler divergence (D)	(210)-(211)	`quick_test_DKL.m`
Rényi divergence ($D_{\mathrm{R},\alpha}$)	(213)	`quick_test_DRenyi.m`
L_2 divergence (D_{L})	(215)	`quick_test_DL2.m`
Jensen-Rényi divergence ($D^{\pi}_{\mathrm{JR},\alpha}$)	(216)	`quick_test_DJensen_Renyi.m`
Bregman distance ($D_{\mathrm{NB},\alpha}$)	(217)	`quick_test_DBregman.m`
χ^2 distance (D_{χ^2})	(218), (219)	`quick_test_DChiSquare.m`
Hellinger distance (D_H)	(220)	`quick_test_DHellinger.m`
Tsallis divergence ($D_{\mathrm{T},\alpha}$)	(214)	`quick_test_DTsallis.m`
cross-entropy (C_{CE})	(221)-(222)	`quick_test_CCE.m`
expected kernel (K_{exp})	(223)	`quick_test_Kexpected.m`
probability product kernel ($K_{\mathrm{PP},\rho}$)	(226)-(227)	`quick_test_KPP.m`
exponentiated Jensen-Rényi kernel-1 ($K_{\mathrm{EJR1},u,\alpha}$)	(228)	`quick_test_KEJR1.m`
exponentiated Jensen-Rényi kernel-2 ($K_{\mathrm{EJR2},u,\alpha}$)	(229)	`quick_test_KEJR2.m`
exponentiated Jensen-Tsallis kernel-1 ($K_{\mathrm{EJT1},u,\alpha}$)	(230)	`quick_test_KEJT1.m`
exponentiated Jensen-Tsallis kernel-2 ($K_{\mathrm{EJT2},u,\alpha}$)	(231)	`quick_test_KEJT2.m`
conditional Shannon entropy [$H(\vert)$]	(163)&(200)	`quick_test_condHShannon.m`
conditional Shannon mutual information [$I(\vert)$]	(164)&(200)	`quick_test_condIShannon.m`

Table 29: Quick tests: analytical formula vs. estimated value. First column: estimated quantity; click on the quantities to see their definitions. Second column: analytical formula. Third column: quick test.

Estimated quantity	Auxiliary quantities	Condition	$\Leftarrow$	Sufficient	`cost_name`
Sharma-Mittal entropy ($H_{\mathrm{SM},\alpha,\beta}$)	$I_\alpha \Leftarrow F$	$k \equiv 0$, $\alpha\boldsymbol{\theta} \in \boldsymbol{\Theta}$		$\boldsymbol{\Theta}$: convex cone[a]	'SharmaM_expF'
Rényi entropy ($H_{\mathrm{R},\alpha}$)	F	$k \equiv 0$, $\alpha\boldsymbol{\theta} \in \boldsymbol{\Theta}$		$\boldsymbol{\Theta}$: convex cone[a]	'Renyi_expF'
Tsallis entropy ($H_{\mathrm{T},\alpha}$)	F	$k \equiv 0$, $\alpha\boldsymbol{\theta} \in \boldsymbol{\Theta}$		$\boldsymbol{\Theta}$: convex cone[a]	'Tsallis_expF'
Shannon entropy (H)	F, ∇F	$k \equiv 0$			'Shannon_expF'
Cross-entropy (C_{CE})	F, ∇F	$k \equiv 0$			'CE_expF'
Sharma-Mittal divergence ($D_{\mathrm{SM},\alpha,\beta}$)	$D_{\mathrm{temp1}} \Leftarrow J_{F,\alpha} \Leftarrow F$	$\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2 \in \boldsymbol{\Theta}$		$\alpha \in (0,1)$	'SharmaM_expF'
Kullback-Leibler divergence ((D)	$B_F \Leftarrow F, \nabla F$				'KL_expF'

Table 30: Summary of analytical expressions for information theoretical quantities in the exponential family. First column: estimated quantity; click on the quantities to see their definitions. Second column: auxiliary variables used for estimation. Third column: condition for the validness of the analytical expressions. Fourth column: sufficient conditions for the third column. Fifth column: name of the estimator in ITE.

[a] This is sufficient for the latter ($\alpha\boldsymbol{\theta} \in \boldsymbol{\Theta}$) condition.

Task	Notation	Quick test
independence of $\mathbf{y}^m$-s $\stackrel{?}{\Rightarrow} I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right)=0$	I: mutual information	`quick_test_Iindependence.m`
$f_1=f_2 \stackrel{?}{\Rightarrow} D\left(f_1,f_2\right)=0$	D: divergence	`quick_test_Dequality.m`
independence of y^m-s $\stackrel{?}{\Rightarrow} A\left(y^1,\ldots,y^M\right)=0$	A: association	`quick_test_Aindependence.m`
$y^1=y^2$ (in distribution/realization) $\stackrel{?}{\Rightarrow} A\left(y^1,y^2\right)=0/1$	A: association	`quick_test_Aequality.m`
$f_1,\ldots,f_M \stackrel{?}{\Rightarrow} G=[G_{ij}]=[K(f_i,f_j)]_{i,j=1}^M$ positive semi-definite	K: kernel	`quick_test_Kpos_semidef.m`

Table 31: Quick tests: independence/equality. First column: task. Second column: notation. Third column: quick test.

Used objective	Quick test
entropy	`quick_test_Himreg.m`
mutual information	`quick_test_Iimreg.m`

Table 32: Quick tests: image registration. First column: used objective. Second column: quick test.

6.2 Further Consistency Tests

This consistency group contains 5 different quick tests, see Table 31. The tests

1. compare the exact values (0 or 1) versus their estimations as a function of the sample number upon independence, or equality of the underlying distributions,
2. focus on the positive semi-definiteness of the Gram matrix associated to a distribution kernel.

6.3 Information Theoretical Image Registration

In image registration one has two images, $\mathbf{I}_{\text{ref}}$ and $\mathbf{I}_{\text{test}}$, as well as a family of geometrical transformations ($\boldsymbol{\theta}\in\boldsymbol{\Theta}$). The goal is to find the transformation (parameter $\boldsymbol{\theta}$) for which the warped test image $\mathbf{I}_{\text{test}}(\boldsymbol{\theta})$ is the 'closest' possible to reference image $\mathbf{I}_{\text{ref}}$:

$$J_{\text{im}}(\boldsymbol{\theta})=S(\mathbf{I}_{\text{ref}},\mathbf{I}_{\text{test}}(\boldsymbol{\theta}))\to\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}},$$

where the similarity of two images is measured by S. Let us also given a feature representation $f(p;\mathbf{I})\in\mathbb{R}^{d_f}$ of image $\mathbf{I}$ in pixel $p\in\mathbf{I}$. These features can be used to solve optimization problem (6.3). In the image registration quick tests of ITE

1. the feature representations (f) are the neighborhoods of the pixels,
2. the geometrical transformations are rotations ($\boldsymbol{\theta}$), and
3. the similarity (S) can be the mutual information, or the joint entropy of the image representations

$$S_{\text{I}}(\mathbf{I}_{\text{ref}},\mathbf{I}_{\text{test}}(\boldsymbol{\theta}))=I\left(\mathbf{I}_{\text{ref}},\mathbf{I}_{\text{test}}(\boldsymbol{\theta})\right),\quad (d_1=d_2=d_f) \tag{255}$$

$$S_{\text{H}}(\mathbf{I}_{\text{ref}},\mathbf{I}_{\text{test}}(\boldsymbol{\theta}))=-H\left([\mathbf{I}_{\text{ref}};\mathbf{I}_{\text{test}}(\boldsymbol{\theta})]\right),\quad (d=2d_f). \tag{256}$$

The corresponding methods are enlisted in Table 32.

6.4 Distribution Regression

In distribution regression an $\mathbf{z}=\{(x_i,y_i)\}_{i=1}^l$ sample is considered, where $y_i\in\mathbb{R}$ and each x_i is a probability distribution on domain $\mathcal{X}$ ($x_i\in\mathcal{M}_+^1(\mathcal{X})$). However, we do not observe x_i directly; rather we observe a sample $x_{i,1},\ldots,x_{i,N_i}\overset{i.i.d.}{\sim}x_i$. Thus, the observed data are $\hat{\mathbf{z}}=\{\{(x_{i,n}\}_{n=1}^{N_i},y_i\}_{i=1}^l$. Our goal is to predict a new y_{l+1} from a new batch of samples $x_{l+1,1},\ldots,x_{l+1,N_{l+1}}$ drawn from a new distribution x_{l+1}, based on $\hat{\mathbf{z}}$.

There is a *two-stage sampling* difficulty in the problem: we only have samples ($\{x_{i,n}\}_{n=1}^{N_i}$) from sampled distributions (x_i). Examples covered by this task family include

Application	Quick test
supervised entropy learning	`quick_test_supervised_entropy_learning.m`
aerosol optical depth prediction	`quick_test_AOD_prediction_linearK.m`, `quick_test_AOD_prediction_nonlinearK`

Table 33: Quick tests: distribution regression. First column: application. Second column: quick test.

1. *bag of features* (BoF) representation based prediction problems (the distributions are represented by bags of instances),
2. *multiple instance learning* approaches (each bag is tagged by one label only) [31, 129, 33].

It has been recently shown [159] that the (x, y) random relation can be learned (statistically consistently) in the two-stage sampled setting by embedding the distributions to a RKHS, followed by a ridge regression optimization. This algorithmically very simple approach

1. theoretically first establishes the applicability of set kernels [54, 43] in the distribution-to-real regression setting,
2. has found new applications [159] in (i) supervised entropy learning [119] and (ii) aerosol optical depth prediction based on satellite images [192]. These MERR (mean embedding based ridge regression) applications are available in ITE, see Table 33.

7 Directory Structure of the Package

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

- *code*: code of ITE,
  - *estimators*: estimators of entropy, mutual information, divergence, association and cross measures, kernels on distributions (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).
    - *quick_ tests*: quick tests to study the efficiency of the estimators, including consistency (analytical vs. estimated value), positive semi-definiteness of the Gram matrices determined by distribution kernels, image registration and distribution regression (supervised entropy learning, aerosol optical depth prediction based on satellite images); see Section 6.
      - *tests_ analytical_ vs_ estimation*, *tests_ other_ consistency*, *tests_ image_ registration*, *tests_ distribution_ regression*: tests grouped into categories.
      - *utilities*: code shared by the quick tests.
    - *utilities*: code shared by *base*, *meta* and *quick_ tests*.
      - *exp_ family*: code associated to analytical information theoretical computations in the exponential family.
      - *analytical_ values*: analytical values of information theoretical quantitites.
      - *MERR*: ingredients of the MERR based distribution regression applications.
  - *IPA*: application of the information theoretical estimators in ITE (see Section 5):
    - *data_ generation*: IPA generators corresponding to different datasets and models.
      - *datasets*: sampling from and plotting of the sources (see Table 27, Table 28, Fig. 5).
      - *models*: IPA model generators, see Table 26.
    - *demos*: IPA demonstrations and estimators, see Table 22 and Table 23.
    - *optimization*: IPA optimization methods (see Table 17, Table 18, Table 19, Table 20, and Table 21).
  - *shared*: code shared by *estimators* and *IPA*.
    - *downloaded*, *embedded*: downloaded and embedded packages (see Section 2).
- *doc*: contains a link to this manual.

A Citing the ITE Toolbox

If you use the ITE toolbox in your work, please cite the paper(s):

- ITE toolbox:

```
@ARTICLE{szabo14information,
  AUTHOR =       {Zolt{\'a}n Szab{\'o}},
  TITLE =        {Information Theoretical Estimators Toolbox},
  JOURNAL =      {Journal of Machine Learning Research},
  YEAR =         {2014},
  volume =       {15},
  pages =        {283-287},
  note =         {(\url{https://bitbucket.org/szzoli/ite/})},
}
```

- ISA separation theorem and its generalizations = basis of the ISA/IPA solvers:

```
@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},
}
```

B Abbreviations

The abbreviations used in the paper are listed in Table 34.

C Functions with Octave-Specific Adaptations

Functions with Octave-specific adaptations are summarized in Table 35.

D Further Definitions

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

Definition 1 (concordance ordering) *In two dimensions ($d = 2$) a C_1 copula is said to be smaller than the C_2 copula ($C_1 \prec C_2$) [98], if*

$$C_1(\mathbf{u}) \leq C_2(\mathbf{u}), \quad \left(\forall \mathbf{u} \in [0,1]^2\right). \tag{257}$$

This pointwise partial ordering on the set of copulas is called concordance ordering.

In the general ($d \geq 2$) case, a C_1 copula is said to be smaller than the C_2 copula ($C_1 \prec C_2$) [67], if

$$C_1(\mathbf{u}) \leq C_2(\mathbf{u}), \text{ and } \bar{C}_1(\mathbf{u}) \leq \bar{C}_2(\mathbf{u}) \quad \left(\forall \mathbf{u} \in [0,1]^d\right). \tag{258}$$

Note:

- *'$\prec$' is called* concordance ordering*; it again defines a partial ordering.*
- *The rationale behind requiring $C_1 \leq C_2$* and *$\bar{C}_1 \leq \bar{C}_2$ is that we want to capture 'simultaneously large'* and *'simultaneously small' tendencies.*

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
BoF	bag of features
BSD	blind source deconvolution
BSSD	blind subspace deconvolution
CDSS	continuously differentiable sample spacing
CE	cross-entropy
EASI	equivariant adaptive separation via independence
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
MERR	mean embedding based ridge regression
MLE	maximum likelihood estimation
MMD	maximum mean discrepancy
NIW	normal-inverted Wishart
NN	nearest neighbor
PCA	principal component analysis
PNL	post nonlinear
PSD	power spectral density
QMI	quadratic mutual information
RBF	radial basis function
RKHS	reproducing kernel Hilbert space
RP	random projection

Table 34: 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 35: Functions with Octave-specific adaptations, i.e, the functions calling `working_environment_Matlab.m` (directly).

- *The two definitions [(257), (258)] coincide only in the two-dimensional ($d = 2$) case.*

Definition 2 (measure of concordance [133, 97, 98]) *A κ 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:*

***A1. Domain**: it is defined for every (y^1, y^2) pair of continuous random variables,*

***A2. Range**: $\kappa\left(y^1, y^2\right) \in [-1, 1]$, [$\kappa\left(y^1, y^1\right) = 1$, and $\kappa\left(y^1, -y^1\right) = -1$],*

***A3. Symmetry**: $\kappa\left(y^1, y^2\right) = \kappa\left(y^2, y^1\right)$,*

***A4. Independence**: if y^1 and y^2 are independent, then $\kappa\left(y^1, y^2\right) = \kappa(\Pi) = 0$,*

***A5. Change of sign**: $\kappa\left(-y^1, y^2\right) = -\kappa\left(y^1, y^2\right)$ [$= \kappa\left(y^1, -y^2\right)$],*

***A6. Coherence**: if $C_1 \prec C_2$, then $\kappa(C_1) \leq \kappa(C_2)$,*[31]

***A7. Continuity**: if $\left(y_t^1, y_t^2\right)$ is a sequence of continuous random variables with copula C_t, and if C_t converges to C pointwise*[32]*, then $\lim_{t\to\infty} \kappa(C_t) = \kappa(C)$.*

Note: properties in the parentheses ('[]') can be derived from the others.

Definition 3 (multivariate measure of concordance [32, 177]) *A* multivariate measure of concordance *is a κ function that assigns to every continuous random variable $\mathbf{y}$ a real number and satisfies the following properties:*

***B1. Normalization**:*

B1a : $\kappa\left(y^1, \ldots, y^d\right) = 1$ if each y^i is an increasing function of every other y^j (or in terms of copulas $\kappa(M) = 1$), and

B1b : $\kappa\left(y^1, \ldots, y^d\right) = 0$ if y^i-s are independent (or in terms of copulas $\kappa(\Pi) = 1$).

***B2. Monotonicity**: $C_1 \prec C_2 \Rightarrow \kappa(C_1) \leq \kappa(C_2)$.*

***B3. Continuity**: If the cdf of the random variable sequence $\mathbf{y}_t$ (F_t) converges to F, the cdf of $\mathbf{y}$ ($\lim_{t\to\infty} F_t = F$), then*

$$\lim_{t\to\infty} \kappa(\mathbf{y}_t) = \kappa(\mathbf{y}). \tag{259}$$

[In terms of copulas: $\lim_{t\to\infty} C_t = C$ (uniformly) $\Rightarrow \lim_{t\to\infty} \kappa(C_t) = \kappa(C)$.]

[31] Hence the name concordance ordering.
[32] In fact uniform convergence of the copulas also holds, see [96] and B3 in Def. 3.

B4. Permutation invariance*: if* $\{i_1, ..., i_d\}$ *is permutation of* $\{1, \ldots, d\}$*, then*

$$\kappa\left(y^{i_1}, \ldots, y^{i_d}\right) = \kappa\left(y^1, \ldots, y^d\right). \tag{260}$$

B5. Duality*:*

$$\kappa\left(-y^1, \ldots, -y^d\right) = \kappa\left(y^1, \ldots, y^d\right). \tag{261}$$

B6. Reflection symmetry property*:*

$$\sum_{\epsilon_1, \ldots, \epsilon_d = \pm 1} \kappa\left(\epsilon_1 y^1, \ldots, \epsilon_d y^d\right) = 0, \tag{262}$$

where the sum is over all the 2^d *possibilities.*

B7. Transition property*: there exists a sequence of* r_d *numbers such that for all* $\mathbf{y}$

$$r_{d-1}\kappa\left(y^2, \ldots, y^d\right) = \kappa\left(y^1, \ldots, y^d\right) + \kappa\left(-y^1, \ldots, y^d\right). \tag{263}$$

Definition 4 (measure of dependence) *[98] defined a numeric measure* κ *between two random variables* y^1 *and* y^2 *whose copula is* C *as a* measure of dependence *if it satisfies the following properties:*

C1. Domain*:* κ *is defined for every* $\left(y^1, y^2\right)$ *pair.*

C2. Symmetry*:* $\kappa\left(y^1, y^2\right) = \kappa\left(y^2, y^1\right)$.

C3. Range*:* $\kappa\left(y^1, y^2\right) \in [0, 1]$.

C4. Independence*:* $\kappa\left(y^1, y^2\right) = 0$ *if and only if* y^1 *and* y^2 *are independent.*

C5. Strictly monotone functional dependence*:* $\kappa\left(y^1, y^2\right) = 1$ *if and only each of* y^1 *and* y^2 *is a strictly monotone function of the other.*

C6. Invariance to strictly monotone functions*: if* f_1 *and* f_2 *are strictly monotone functions, then*

$$\kappa\left(y^1, y^2\right) = \kappa\left(f_1(y^1), f_2(y^2)\right). \tag{264}$$

C7. Continuity*: if* $\left(y_t^1, y_t^2\right)$ *is a sequence of random variables with copula* C_n*, and if* $\lim_{t\to\infty} C_t = C$ *(pointwise), then*

$$lim_{t\to\infty}\kappa(C_t) = \kappa(C). \tag{265}$$

Definition 5 (multivariate measure of dependence) *[194] defined the notion of measure of dependence in case of* d *dimension as follows. A* κ *real-valued function is called a* measure of dependence *if it satisfies the properties:*

D1. Domain*:* κ *is defined for any continuously distributed* $\mathbf{y}$*,*

D2. Permutation invariance*: if* $\{i_1, ..., i_d\}$ *is permutation of* $\{1, \ldots, d\}$*, then*

$$\kappa\left(y^{i_1}, \ldots, y^{i_d}\right) = \kappa\left(y^1, \ldots, y^d\right). \tag{266}$$

D3. Normalization*:* $0 \leq \kappa\left(y^1, \ldots, y^d\right) \leq 1$.

D4. Independence*:* $\kappa\left(y^1, \ldots, y^d\right) = 0$ *if and only if* y^i*-s are independent.*

D5. Strictly monotone functional dependence*:* $\kappa\left(y^1, \ldots, y^d\right) = 1$ *if and only if each* y^i *is an increasing function of each of the others.*

D6. Invariance to strictly monotone functions*: If* $f_1, \ldots, f_d$ *are all strictly increasing functions, then*

$$\kappa\left(y^1, \ldots, y^d\right) = \kappa\left(f_1\left(y^1\right), \ldots, f_d\left(y^d\right)\right). \tag{267}$$

***D7. Normal case**: Let* $\mathbf{y}$ *be normally distributed and* $\rho_{ij} = cov\left(y^i, y^j\right)$*. If* r_{ij}*-s are either all non-negative, or all non-positive then* κ *is a strictly increasing function of each of the* $|r_{ij}|$*-s.*

***D8. Continuity**: If the random variable sequence* $\mathbf{y}_t$ *converges in distribution to* $\mathbf{y}$*, then*

$$\lim_{t\to\infty} \kappa(\mathbf{y}_t) = \kappa(\mathbf{y}). \tag{268}$$

Definition 6 (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, ρ) *is called a* semimetric space.[33] *A semimetric space is said to be of* negative type *if*

$$\sum_{i=1}^{T}\sum_{j=1}^{T} a_i a_j \rho(z_i, z_j) \leq 0 \tag{269}$$

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* $q \in (0, 2]$*.*

Definition 7 ((covariant) Hilbertian metric) *A* $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *metric is Hilbertian if there exist a Hilbert space* $\mathcal{H}$ *and an* $f : \mathcal{X} \to \mathcal{H}$ *isometry [56]:*

$$\rho^2(x, y) = \langle f(x) - f(y), f(x) - f(y)\rangle_{\mathcal{H}} = \|f(x) - f(y)\|_{\mathcal{H}}^2 \quad (\forall x, y \in \mathcal{X}). \tag{270}$$

Additionally, if $\mathcal{X}$ *is the set of distributions (*$\boldsymbol{\mathcal{M}}_+^1(\mathcal{X})$*) and* ρ *is independent of the dominating measure, then* d *is called* covariant*. Intuitively, this means its value is invariant to arbitrary smooth coordinate transformations of the underlying probability space; for example, it is no matter whether we take RGB, HSV, ... color space.*

E Estimation Formulas – Lookup Table

In this section the computations of entropy (Section E.1), mutual information (Section E.2), divergence (Section E.3), association (Section E.4) and cross (Section E.5) measures, and kernels on distributions (Section E.6) 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), exponential, Cauchy, generalized t-student, polynomial kernel, rational quadratic, inverse multiquadratic and Matérn kernels (with $\frac{3}{2}$ and $\frac{5}{2}$ smoothness parameters) are defined as

$$k_G(\mathbf{u}, \mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}}, \qquad k_e(\mathbf{u}, \mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2}{2\sigma^2}}, \tag{271}$$

$$k_C(\mathbf{u}, \mathbf{v}) = \frac{1}{1 + \frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{\sigma^2}}, \qquad k_t(\mathbf{u}, \mathbf{v}) = \frac{1}{1 + \|\mathbf{u} - \mathbf{v}\|_2^t}, \tag{272}$$

$$k_p(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v}\rangle + c)^p, \qquad k_r(\mathbf{u}, \mathbf{v}) = 1 - \frac{\|\mathbf{u} - \mathbf{v}\|_2^2}{\|\mathbf{u} - \mathbf{v}\|_2^2 + c}, \tag{273}$$

$$k_i(\mathbf{u}, \mathbf{v}) = \frac{1}{\sqrt{\|\mathbf{u} - \mathbf{v}\|_2^2 + c^2}}, \tag{274}$$

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

$$k_{M,\frac{3}{2}}(\mathbf{u},\mathbf{v})=\left(1+\frac{\sqrt{3}\left\|\mathbf{u}-\mathbf{v}\right\|_2}{l}\right)e^{-\frac{\sqrt{3}\|\mathbf{u}-\mathbf{v}\|_2}{l}},\quad k_{M,\frac{5}{2}}(\mathbf{u},\mathbf{v})=\left(1+\frac{\sqrt{5}\left\|\mathbf{u}-\mathbf{v}\right\|_2}{l}+\frac{5\left\|\mathbf{u}-\mathbf{v}\right\|_2^2}{3l^2}\right)e^{-\frac{\sqrt{5}\|\mathbf{u}-\mathbf{v}\|_2}{l}},\tag{275}$$

where $p\in\mathbb{Z}^+$ and $\sigma>0,t>0,l>0,c>0$. $tr(\cdot)$ stands for trace. Let $N(\mathbf{m},\mathbf{\Sigma})$ denote the density function of the normal random variable with mean $\mathbf{m}$ and covariance $\mathbf{\Sigma}$.

$$V_d=\frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}=\frac{2\pi^{d/2}}{d\Gamma\left(\frac{d}{2}\right)}\tag{276}$$

is the volume of the d-dimensional unit ball. ψ is the digamma function. Let

$$I_\alpha=\int_{\mathbb{R}^d}\left[f(\mathbf{u})\right]^\alpha\mathrm{d}\mathbf{u},\tag{277}$$

where f is a probability density on $\mathbb{R}^d$. The inner 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. $\mathbb{E}$ is the expectation operator.

- `Shannon_kNN_k` [75, 147, 45]:

$$\hat{H}(\mathbf{Y}_{1:T})=\log(T-1)-\psi(k)+\log(V_d)+\frac{d}{T}\sum_{t=1}^{T}\log\left(\rho_k(t)\right).\tag{278}$$

- `Renyi_kNN_k` [196, 81]:

$$C_{\alpha,k}=\left[\frac{\Gamma(k)}{\Gamma(k+1-\alpha)}\right]^{\frac{1}{1-\alpha}},\tag{279}$$

$$\hat{I}_\alpha(\mathbf{Y}_{1:T})=\frac{T-1}{T}V_d^{1-\alpha}C_{\alpha,k}^{1-\alpha}\sum_{t=1}^{T}\frac{[\rho_k(t)]^{d(1-\alpha)}}{(T-1)^\alpha},\tag{280}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T})=\frac{1}{1-\alpha}\log\left(\hat{I}_\alpha(\mathbf{Y}_{1:T})\right),\tag{281}$$

or equivalently (see `Renyi_kNN_S`)

$$S=\{k\},\tag{282}$$

$$V=\mathbf{Y}_{1:T},\tag{283}$$

$$L(V)=\sum_{(\mathbf{u},\mathbf{v})\in edges(NN_S(V))}\left\|\mathbf{u}-\mathbf{v}\right\|_2^{d(1-\alpha)},\tag{284}$$

$$c=\lim_{T\to\infty}\mathbb{E}_{U_{1:T},u_t:i.i.d.,\sim Uniform([0,1]^d)}\left[\frac{L(U_{1:T})}{T^\alpha}\right],\tag{285}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T})=\frac{1}{1-\alpha}\log\left[\frac{L(V)}{cT^\alpha}\right].\tag{286}$$

Making Eq. (281) and Eq. (286) equal, one gets $\frac{T-1}{T}(V_dC_{\alpha,k})^{1-\alpha}\frac{1}{(T-1)^\alpha}=\frac{1}{cT^\alpha}\Rightarrow c=\left(\frac{T-1}{T}\right)^{\alpha-1}(V_dC_{\alpha,k})^{\alpha-1}=\left(\frac{T-1}{T}V_d\right)^{\alpha-1}\frac{\Gamma(k+1-\alpha)}{\Gamma(k)}=:c_k$ using the definition of $C_{\alpha,k}$ [Eq. (279)].

- `Renyi_kNN_1tok` [117]:

$$S = \{1, \ldots, k\}, \tag{287}$$
$$V = \mathbf{Y}_{1:T}, \tag{288}$$
$$L(V) = \sum_{(\mathbf{u},\mathbf{v}) \in edges(NN_S(V))} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{289}$$
$$c_j = \left(\frac{T-1}{T} V_d\right)^{\alpha-1} \frac{\Gamma(k+1-\alpha)}{\Gamma(k)}, \tag{290}$$
$$c = \lim_{T\to\infty} \mathbb{E}_{U_{1:T}, u_t: i.i.d., \sim Uniform([0,1]^d)} \left[\frac{L(U_{1:T})}{T^\alpha}\right] = \sum_{j=1}^{k} c_j, \tag{291}$$
$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log\left[\frac{L(V)}{cT^\alpha}\right]. \tag{292}$$

- `Renyi_kNN_S` [110]:

$$S \subseteq \{1, \ldots, k\}, k \in S, \tag{293}$$
$$V = \mathbf{Y}_{1:T}, \tag{294}$$
$$L(V) = \sum_{(\mathbf{u},\mathbf{v}) \in edges(NN_S(V))} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{295}$$
$$c_j = \left(\frac{T-1}{T} V_d\right)^{\alpha-1} \frac{\Gamma(k+1-\alpha)}{\Gamma(k)}, \tag{296}$$
$$c = \lim_{T\to\infty} \mathbb{E}_{U_{1:T}, u_t: i.i.d., \sim Uniform([0,1]^d)} \left[\frac{L(U_{1:T})}{T^\alpha}\right] = \sum_{j\in S} c_j, \tag{297}$$
$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log\left[\frac{L(V)}{cT^\alpha}\right]. \tag{298}$$

- `Renyi_weightedkNN` [151]:

$$k_1 = k_1(T) = \left\lceil 0.1\sqrt{T} \right\rceil, \tag{299}$$
$$k_2 = k_2(T) = \left\lceil 2\sqrt{T} \right\rceil, \tag{300}$$
$$N = \left\lfloor \frac{T}{2} \right\rfloor \tag{301}$$
$$M = T - N, \tag{302}$$
$$V_1 = Y_{1:N}, \tag{303}$$
$$V_2 = Y_{N+1:T}, \tag{304}$$
$$S = \{k_1, \ldots, k_2\}, \tag{305}$$
$$\eta_k = \frac{\beta(k, 1-\alpha)}{\Gamma(1-\alpha)} \frac{1}{N} M^{1-\alpha} V_d^{1-\alpha} \sum_{(\mathbf{u},\mathbf{v}) \in edges(NN_S(V_2, V_1))} \|\mathbf{u} - \mathbf{v}\|_2^{d(1-\alpha)}, \tag{306}$$
$$\hat{I}_{\alpha,\mathbf{w}} = \sum_{k\in S} w_k \eta_k, \tag{307}$$
$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log(\hat{I}_{\alpha,\mathbf{w}}), \tag{308}$$

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

- `Renyi_MST` [196]:

$$V = \mathbf{Y}_{1:T}, \tag{309}$$

$$L(V) = \min_{G\in \text{ spanning trees on } V} \sum_{(\mathbf{u},\mathbf{v})\in edges(G)} \|\mathbf{u}-\mathbf{v}\|_2^{d(1-\alpha)}, \tag{310}$$

$$c = \lim_{T\to\infty} \mathbb{E}_{U_{1:T},u_t:i.i.d.,\sim Uniform([0,1]^d)} \left[\frac{L(U_{1:T})}{T^\alpha}\right], \tag{311}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha}\log\left[\frac{L(V)}{cT^\alpha}\right]. \tag{312}$$

- `Tsallis_kNN_k` [81]:

$$C_{\alpha,k} = \left[\frac{\Gamma(k)}{\Gamma(k+1-\alpha)}\right]^{\frac{1}{1-\alpha}}, \tag{313}$$

$$\hat{I}_\alpha(\mathbf{Y}_{1:T}) = \frac{T-1}{T} V_d^{1-\alpha} C_{\alpha,k}^{1-\alpha} \sum_{t=1}^{T} \frac{[\rho_k(t)]^{d(1-\alpha)}}{(T-1)^\alpha}, \tag{314}$$

$$\hat{H}_{\mathrm{T},\alpha}(\mathbf{Y}_{1:T}) = \frac{1-\hat{I}_\alpha(\mathbf{Y}_{1:T})}{\alpha-1}. \tag{315}$$

- `Shannon_Edgeworth` [60]: 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

$$\hat{H}(\mathbf{Y}_{1:T}) = H(\phi_d) - \frac{1}{12}\left[\sum_{i=1}^{d}\left(\kappa^{i,i,i}\right)^2 + 3\sum_{i,j=1;i\neq j}^{d}\left(\kappa^{i,i,j}\right)^2 + \frac{1}{6}\sum_{i,j,k=1;i<j<k}^{d}\left(\kappa^{i,j,k}\right)^2\right], \tag{316}$$

  where

$$\mathbf{y}_t = \mathbf{y}_t - \frac{1}{T}\sum_{k=1}^{T}\mathbf{y}_k, (t=1,\ldots,T) \tag{317}$$

$$\mathbf{\Sigma} = \hat{cov}(\mathbf{Y}_{1:T}) = \frac{1}{T-1}\sum_{t=1}^{T}\mathbf{y}_t(\mathbf{y}_t)^*, \tag{318}$$

$$H(\phi_d) = \frac{1}{2}\log\det(\mathbf{\Sigma}) + \frac{d}{2}\log(2\pi) + \frac{d}{2}, \tag{319}$$

$$\sigma_i = \hat{std}(y^i) = \frac{1}{T-1}\sum_{t=1}^{T}\left(y_t^i\right)^2, \quad (i=1,\ldots,d) \tag{320}$$

$$\kappa^{ijk} = \hat{E}\left[y^i y^j y^k\right] = \frac{1}{T}\sum_{t=1}^{T} y_t^i y_t^j y_t^k, \quad (i,j,k=1,\ldots,d) \tag{321}$$

$$\kappa^{i,j,k} = \frac{\kappa^{ijk}}{\sigma_i\sigma_j\sigma_k}. \tag{322}$$

- `Shannon_Voronoi` [91]: 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:

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T-K}\sum_{V_i:vol(V_i)\neq\infty}\log\left[T\times vol(V_i)\right], \tag{323}$$

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

- `Shannon_spacing_V` [185]:

$$m = m(T) = \left\lfloor\sqrt{T}\right\rfloor, \tag{324}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\log\left(\frac{T}{2m}\left[y_{(i+m)} - y_{(i-m)}\right]\right). \tag{325}$$

- `Shannon_spacing_Vb` [184]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{326}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T-m} \sum_{t=1}^{T-m} \log \left[\frac{T+1}{m} (y_{(t+m)} - y_{(t)}) \right] + \sum_{k=m}^{T} \frac{1}{k} + \log \left(\frac{m}{T+1} \right). \tag{327}$$

- `Shannon_spacing_Vpconst` [106]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{328}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T} \sum_{t=1}^{T} \log \left[\frac{T}{c_t m} (y_{(t+m)} - y_{(t-m)}) \right], \tag{329}$$

where

$$c_t = \begin{cases} 1, & 1 \le t \le m, \\ 2, & m+1 \le t \le T-m, \\ 1 & T-m+1 \le t \le T. \end{cases} \tag{330}$$

It can be shown [106] that (325) = (329)+$\frac{2m \log(2)}{T}$.

- `Shannon_spacing_Vplin` [34]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{331}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T} \sum_{t=1}^{T} \log \left[\frac{T}{c_t m} (y_{(t+m)} - y_{(t-m)}) \right], \tag{332}$$

where

$$c_t = \begin{cases} 1 + \frac{t-1}{m}, & 1 \le t \le m, \\ 2, & m+1 \le t \le T-m, \\ 1 + \frac{T-t}{m} & T-m+1 \le t \le T. \end{cases} \tag{333}$$

- `Shannon_spacing_Vplin2` [34]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{334}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \frac{1}{T} \sum_{t=1}^{T} \log \left[\frac{T}{c_t m} (y_{(t+m)} - y_{(t-m)}) \right], \tag{335}$$

$$\tag{336}$$

where

$$c_t = \begin{cases} 1 + \frac{t+1}{m} - \frac{t}{m^2}, & 1 \le t \le m, \\ 2, & m+1 \le t \le T-m-1, \\ 1 + \frac{T-t}{m+1}, & T-m \le t \le T. \end{cases} \tag{337}$$

- `Shannon_spacing_LL` [23]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{338}$$

$$\bar{y}_{(i)} = \frac{1}{2m+1} \sum_{j=i-m}^{i+m} y_{(j)}, \tag{339}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = -\frac{1}{T} \sum_{t=1}^{T} \log \left[\frac{\sum_{j=i-m}^{i+m} \left(y_{(j)} - \bar{y}_{(i)} \right) (j-i)}{T \sum_{j=i-m}^{i+m} \left(y_{(j)} - \bar{y}_{(i)} \right)^2} \right]. \tag{340}$$

- `Renyi_spacing_V` [188]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{341}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log \left[\frac{1}{T} \sum_{t=1}^{T} \left(\frac{T}{2m} \left[y_{(i+m)} - y_{(i-m)} \right] \right)^{1-\alpha} \right]. \tag{342}$$

- `Renyi_spacing_E` [188]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{343}$$

$$t_1 = \sum_{i=2-m}^{0} \frac{y_{(i+m)} - y_{(i+m-1)}}{2} \left(\sum_{j=1}^{i+m-1} \frac{2}{y_{(j+m)} - y_{(j-m)}} \right)^{\alpha}, \tag{344}$$

$$t_2 = \sum_{i=1}^{T+1-m} \frac{y_{(i)} + y_{(i+m)} - y_{(i-1)} - y_{(i+m-1)}}{2} \left(\sum_{j=i}^{i+m-1} \frac{2}{y_{(j+m)} - y_{(j-m)}} \right)^{\alpha}, \tag{345}$$

$$t_3 = \sum_{i=T+2-m}^{T} \frac{y_{(i)} - y_{(i-1)}}{2} \left(\sum_{j=i}^{T} \frac{2}{y_{(j+m)} - y_{(j-m)}} \right)^{\alpha}, \tag{346}$$

$$\hat{H}_{\mathrm{R},\alpha}(\mathbf{Y}_{1:T}) = \frac{1}{1-\alpha} \log \left[\frac{t_1 + t_2 + t_3}{T^{\alpha}} \right]. \tag{347}$$

- `qRenyi_CDSS` [108]:

$$m = m(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{348}$$

$$\hat{H}_{\mathrm{R},2}(\mathbf{Y}_{1:T}) = -\log \left[\frac{30}{T(T-m)} \sum_{i=1}^{T-m} \sum_{j=i+1}^{i+m-1} \frac{(y_{(j)} - y_{(i+m)})^2 (y_{(j)} - y_{(i)})^2}{(y_{(i+m)} - y_{(i)})^5} \right]. \tag{349}$$

- `Shannon_KDP` [153]:

$$\text{adaptive (k-d) partitioning} \Rightarrow \mathcal{A} = \{A_1, \ldots, A_K\}, \tag{350}$$

$$T_k = \#\left\{ t : 1 \le t \le T, \mathbf{y}_t \in A_k \right\}, \tag{351}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = \sum_{k=1}^{K} \frac{T_k}{T} \log \left[\frac{T}{T_k} vol(A_k) \right], \tag{352}$$

where 'vol' denotes volume.

- `Shannon_MaxEnt1`, `Shannon_MaxEnt2` [25, 61]: Since the Shannon entropy is invariant to additive constants ($H(y) = H(y+m)$), one can assume without loss of generality that the expectation of y is zero. The maximum entropy distribution based entropy estimators (assuming y with unit standard deviation) take the form

$$H(n) - \left[k_1 \mathbb{E}^2 \left[G_1(y) \right] + k_2 \left(\mathbb{E}\left[G_2(y) \right] - \mathbb{E}\left[G_2(n) \right] \right)^2 \right] \tag{353}$$

with suitably chosen $k_i \in \mathbb{R}$ contants and G_i functions ($i = 1, 2$). In Eq. (353),

$$H(n) = \frac{1 + \log(2\pi)}{2} \tag{354}$$

denotes the entropy of the standard normal variable (n), and in practise expectations are changed to their empirical variants. Particularly,

– `Shannon_MaxEnt1`:

$$\hat{\sigma} = \hat{\sigma}\left(\mathbf{Y}_{1:T}\right) = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(y_t\right)^2}, \tag{355}$$

$$y'_t = y_t/\hat{\sigma}, \quad (t = 1, \ldots, T) \tag{356}$$

$$G_1(z) = ze^{\frac{-z^2}{2}}, \tag{357}$$

$$G_2(z) = |z|, \tag{358}$$

$$k_1 = \frac{36}{8\sqrt{3}-9}, \tag{359}$$

$$k_2 = \frac{1}{2-\frac{6}{\pi}}, \tag{360}$$

$$\hat{H}_0 = \hat{H}_0\left(\mathbf{Y}'_{1:T}\right) = H(n) - \left[k_1\left(\frac{1}{T}\sum_{t=1}^{T}G_1\left(y'_t\right)\right)^2 + k_2\left(\frac{1}{T}\sum_{t=1}^{T}G_2\left(y'_t\right) - \sqrt{\frac{2}{\pi}}\right)^2\right], \tag{361}$$

$$\hat{H}\left(\mathbf{Y}_{1:T}\right) = \hat{H}_0 + \log(\hat{\sigma}). \tag{362}$$

– `Shannon_MaxEnt2`:

$$\hat{\sigma} = \hat{\sigma}\left(\mathbf{Y}_{1:T}\right) = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(y_t\right)^2}, \tag{363}$$

$$y'_t = y_t/\hat{\sigma}, \quad (t = 1, \ldots, T) \tag{364}$$

$$G_1(z) = ze^{\frac{-z^2}{2}}, \tag{365}$$

$$G_2(z) = e^{\frac{-z^2}{2}}, \tag{366}$$

$$k_1 = \frac{36}{8\sqrt{3}-9}, \tag{367}$$

$$k_2 = \frac{24}{16\sqrt{3}-27}, \tag{368}$$

$$\hat{H}_0 = \hat{H}_0\left(\mathbf{Y}'_{1:T}\right) = H(n) - \left[k_1\left(\frac{1}{T}\sum_{t=1}^{T}G_1\left(y'_t\right)\right)^2 + k_2\left(\frac{1}{T}\sum_{t=1}^{T}G_2\left(y'_t\right) - \frac{1}{\sqrt{2}}\right)^2\right], \tag{369}$$

$$\hat{H}\left(\mathbf{Y}_{1:T}\right) = \hat{H}_0 + \log(\hat{\sigma}). \tag{370}$$

- `Phi_spacing` [184]:

$$m = m(T) = \left\lfloor\sqrt{T}\right\rfloor, \tag{371}$$

$$\hat{H}_{\Phi,w}(\mathbf{Y}_{1:T}) = \frac{1}{2}\frac{1}{T-m}\sum_{j=1}^{T-m}\Phi\left(\frac{m}{T+1}\frac{1}{y_{(j+m)}-y_{(j)}}\right)\left[w(y_{(j)}) + w(y_{j+m})\right]. \tag{372}$$

- `SharmaM_kNN_k` [196, 81] ($\hat{I}_\alpha$):

$$C_{\alpha,k} = \left[\frac{\Gamma(k)}{\Gamma(k+1-\alpha)}\right]^{\frac{1}{1-\alpha}}, \tag{373}$$

$$\hat{I}_\alpha(\mathbf{Y}_{1:T}) = \frac{T-1}{T}V_d^{1-\alpha}C_{\alpha,k}^{1-\alpha}\sum_{t=1}^{T}\frac{[\rho_k(t)]^{d(1-\alpha)}}{(T-1)^\alpha}, \tag{374}$$

$$\hat{H}_{\mathrm{SM},\alpha,\beta}(\mathbf{Y}_{1:T}) = \frac{1}{1-\beta}\left(\left[\hat{I}_\alpha(\mathbf{Y}_{1:T})\right]^{\frac{1-\beta}{1-\alpha}} - 1\right). \tag{375}$$

- `SharmaM_expF` [103]:

$$\hat{\mathbf{m}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t, \tag{376}$$

$$\hat{C} = \frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{y}_t - \hat{\mathbf{m}}\right)\left(\mathbf{y}_t - \hat{\mathbf{m}}\right)^*, \tag{377}$$

$$\hat{\boldsymbol{\theta}} = \left[\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2\right] = \left[\hat{\mathbf{C}}^{-1}\hat{\mathbf{m}}, \frac{1}{2}\hat{\mathbf{C}}^{-1}\right], \tag{378}$$

$$\hat{I}_\alpha(\mathbf{Y}_{1:T}) = e^{F(\alpha\hat{\boldsymbol{\theta}})-\alpha F(\hat{\boldsymbol{\theta}})}, \tag{379}$$

$$\hat{H}_{\mathrm{SM},\alpha,\beta}(\mathbf{Y}_{1:T}) = \frac{1}{1-\beta}\left(\left[\hat{I}_\alpha(\mathbf{Y}_{1:T})\right]^{\frac{1-\beta}{1-\alpha}} - 1\right), \tag{380}$$

where for function F (in case of Gaussian variables), see Eq. (235).

- `Shannon_PSD_SzegoT` [47, 46, 127]: The method assumes that $y \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ (this is assured for the samples by the $y_t \to \frac{y_t}{2a}$ pre-processing, where $a = \max_{t=1,\dots,T}|y_t|$). Let $\lambda_{K,k}$ denote the eigenvalues of the empiricial estimation of the $K \times K$ sized Toeplitz matrix

$$\boldsymbol{\Phi}_K^{(f)} = \left[\int_{-\frac{1}{2}}^{\frac{1}{2}} f(u)e^{i2\pi u(a-b)}\mathrm{d}u\right]_{a,b=1}^{K} = \left[\mathbb{E}\left(e^{i2\pi y(a-b)}\right)\right]_{a,b=1}^{K}, \tag{381}$$

i.e., of

$$\hat{\boldsymbol{\Phi}}_K^{(f)} = \left[\frac{1}{T}\sum_{t=1}^{T} e^{i2\pi y_t(a-b)}\right]_{a,b=1}^{K}, \tag{382}$$

where f is the density of y. The estimator realizes the relation

$$\hat{H}\left(Y_{1:T}\right) = -\frac{1}{K}\sum_{k=1}^{K}\lambda_{K,k}\log\left(\lambda_{K,k}\right). \tag{383}$$

- `Shannon_expF` [102]: MLE estimation ($\hat{\boldsymbol{\theta}}$) plugged into (244).
- `Renyi_expF`, `Tsallis_expF` [102, 103]: MLE estimation ($\hat{\boldsymbol{\theta}}$) plugged into (242) and (243), respectively.
- `Shannon_spacing_VKDE` [107]:

$$m = m(T) = \left\lfloor\sqrt{T}\right\rfloor, \tag{384}$$

$$\hat{H}(\mathbf{Y}_{1:T}) = -\frac{1}{T}\sum_{t=1}^{T}\log s_t(T,m), \tag{385}$$

$$\tag{386}$$

where

$$s_t(T,m) = \begin{cases} \hat{f}(y_{(t)}), & 1 \le t \le m, \\ \frac{2\frac{m}{T}}{y_{(t+m)}-y_{(t-m)}}, & m+1 \le t \le T-m, , \\ \hat{f}(y_{(t)}), & T-m+1 \le t \le T. \end{cases} \tag{387}$$

$$\hat{f}(y) = \frac{1}{Th}\sum_{t=1}^{T} k\left(\frac{y-y_j}{h}\right), \tag{388}$$

$$h = 1.06\hat{\sigma}T^{-\frac{1}{5}}, \tag{389}$$

k is the density of the standard normal variable, and $\hat{\sigma} = \hat{\sigma}\left(\mathbf{Y}_{1:T}\right)$ is the sample standard deviation.

- `Shannon_vME` [72]: $T := H$, ψ is T's influence function, K is a smoothing kernel constructed from Legendre polynomials

$$\hat{f}(\mathbf{y}) = \frac{1}{Th^d} \sum_{t=1}^{T} K\left(\frac{\mathbf{y} - \mathbf{y}_t}{h}\right), \tag{390}$$

$$\hat{T} = T\left(\hat{f}\right) + \frac{1}{T} \sum_{t=1}^{T} \psi\left(y_t; \hat{f}\right). \tag{391}$$

E.2 Mutual Information

Notations. 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:

$$\hat{F}_m(y) = \sum_{t=1}^{T} \mathbb{I}_{\{y_t^m \leq y\}}, \tag{392}$$

let the vector of *grades* be defined as

$$\mathbf{U} = \left[F_1\left(y^1\right); \ldots; F_d\left(y^d\right)\right] \in [0,1]^d, \tag{393}$$

and let its empirical analog, the *ranks* be

$$\hat{U}_{mt} = \hat{F}_m(y_t^m) = \frac{1}{T}(\text{rank of } y_t^m \text{ in } y_1^m, \ldots, y_T^m), \quad (m = 1, \ldots, d). \tag{394}$$

Finally, the empirical copula is defined as

$$\hat{C}_T(\mathbf{u}) := \frac{1}{T} \sum_{t=1}^{T} \prod_{i=1}^{d} \mathbb{I}_{\{\hat{U}_{it} \leq u_i\}}, \quad \left(\mathbf{u} = [u_1; \ldots; u_d] \in [0,1]^d\right), \tag{395}$$

specifically

$$\hat{C}_T\left(\frac{i_1}{T}, \ldots, \frac{i_T}{T}\right) = \frac{\#\text{ of } \mathbf{y}\text{-s in the sample with } \mathbf{y} \leq \mathbf{y}_{(i_1,\ldots,i_T)}}{T}, \quad (\forall j, i_j = 1, \ldots, T) \tag{396}$$

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` [51]:

$$H_{ij} = \delta_{ij} - \frac{1}{T}, \tag{397}$$

$$(\mathbf{K}_m)_{ij} = k_m\left(\mathbf{y}_i^m, \mathbf{y}_j^m\right), \tag{398}$$

$$\hat{I}_{\text{HSIC}}\left(\mathbf{Y}_{1:T}\right) = \frac{1}{T^2} \sum_{u=1}^{M-1} \sum_{v=u+1}^{M} tr(\mathbf{K}_u \mathbf{H} \mathbf{K}_v \mathbf{H}). \tag{399}$$

  Currently, k_m-s are RBF-s.

- `KCCA`, `KGV` [7, 167]:

$$\kappa_2 = \frac{\kappa T}{2}, \tag{400}$$

$$\mathbf{K}_m = \left[k_m(\mathbf{y}_i^m, \mathbf{y}_j^m)\right]_{i,j=1,\ldots,T}, \tag{401}$$

$$\mathbf{H} = \mathbf{I} - \frac{1}{T}\mathbf{1}\mathbf{1}^*, \tag{402}$$

$$\widetilde{\mathbf{K}}_m = \mathbf{H}\mathbf{K}_m\mathbf{H}, \tag{403}$$

$$\begin{pmatrix} (\widetilde{\mathbf{K}}_1+\kappa_2\mathbf{I}_T)^2 & \widetilde{\mathbf{K}}_1\widetilde{\mathbf{K}}_2 & \cdots & \widetilde{\mathbf{K}}_1\widetilde{\mathbf{K}}_M \\ \widetilde{\mathbf{K}}_2\widetilde{\mathbf{K}}_1 & (\widetilde{\mathbf{K}}_2+\kappa_2\mathbf{I}_T)^2 & \cdots & \widetilde{\mathbf{K}}_2\widetilde{\mathbf{K}}_M \\ \vdots & \vdots & & \vdots \\ \widetilde{\mathbf{K}}_M\widetilde{\mathbf{K}}_1 & \widetilde{\mathbf{K}}_M\widetilde{\mathbf{K}}_2 & \cdots & (\widetilde{\mathbf{K}}_M+\kappa_2\mathbf{I}_T)^2 \end{pmatrix} \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_M \end{pmatrix} = \tag{404}$$

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

Let us write Eq. (404) shortly as $\mathbf{Ac} = \lambda\mathbf{Bc}$. Let the minimal eigenvalue of this generalized eigenvalue problem be λ_{KCCA}, and $\lambda_{\text{KGV}} = \frac{\det(\mathbf{A})}{\det(\mathbf{B})}$.

$$\hat{I}_{\text{KCCA}}\left(\mathbf{Y}_{1:T}\right) = -\frac{1}{2}\log(\lambda_{\text{KCCA}}), \tag{405}$$

$$\hat{I}_{\text{KGV}}\left(\mathbf{Y}_{1:T}\right) = -\frac{1}{2}\log(\lambda_{\text{KGV}}). \tag{406}$$

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

- `Hoeffding` [58, 42]: The estimation can be computed as

$$h_2(d) = \left(\frac{2}{(d+1)(d+2)} - \frac{1}{2^d}\frac{d!}{\prod_{i=0}^{d}\left(i+\frac{1}{2}\right)} + \frac{1}{3^d}\right)^{-1}, \tag{407}$$

$$\hat{I}_{\Phi}\left(\mathbf{Y}_{1:T}\right) = \sqrt{h_2(d)\left\{\frac{1}{T^2}\sum_{j=1}^{T}\sum_{k=1}^{T}\prod_{i=1}^{d}\left[1-\max(\hat{U}_{ij},\hat{U}_{ik})\right] - \frac{2}{T}\frac{1}{2^d}\sum_{j=1}^{T}\prod_{i=1}^{d}\left(1-\hat{U}_{ij}^2\right) + \frac{1}{3^d}\right\}}. \tag{408}$$

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

$$h_2(d,T)^{-1} = \frac{1}{T^2}\sum_{j=1}^{T}\sum_{k=1}^{T}\left[1-\max\left(\frac{j}{T},\frac{k}{T}\right)\right]^d - \frac{2}{T}\sum_{j=1}^{T}\left[\frac{T(T-1)-j(j-1)}{2T^2}\right]^d + \frac{1}{3^d}\left[\frac{(T-1)(2T-1)}{2T^2}\right]^d, \tag{409}$$

$$\hat{I}_{\Phi}\left(\mathbf{Y}_{1:T}\right) = \sqrt{h_2(d,T)(t_1-t_2+t_3)}, \tag{410}$$

where

$$t_1 = \frac{1}{T^2}\sum_{j=1}^{T}\sum_{k=1}^{T}\prod_{i=1}^{d}\left[1-\max(\hat{U}_{ij},\hat{U}_{ik})\right], \quad t_2 = \frac{2}{T}\frac{1}{2^d}\sum_{j=1}^{T}\prod_{i=1}^{d}\left(1-\hat{U}_{ij}^2-\frac{1-\hat{U}_{ij}}{T}\right), \quad t_3 = \frac{1}{3^d}\left[\frac{(T-1)(2T-1)}{2T^2}\right]^d. \tag{411}$$

- `SW1`, `SWinf` [138, 194, 74]:

$$\hat{I}_{\text{SW1}}\left(\mathbf{Y}_{1:T}\right) = \hat{\sigma} = 12\frac{1}{T^2-1}\sum_{i_1=1}^{T}\sum_{i_2=1}^{T}\left|\hat{C}_T\left(\frac{i_1}{T},\frac{i_2}{T}\right) - \frac{i_1}{T}\frac{i_2}{T}\right|. \tag{412}$$

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

- `QMI_CS_KDE_direct`, `QMI_CS_KDE_iChol`, `QMI_ED_KDE_iChol` [142]:

$$I_{\text{QMI-CS}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \log\left[\frac{L_1L_2}{(L_3)^2}\right], \tag{413}$$

$$I_{\text{QMI-ED}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = L_1+L_2-2L_3, \tag{414}$$

$$(\mathbf{K}_m)_{ij} = k_m(\mathbf{y}_i^m,\mathbf{y}_j^m), \tag{415}$$

$$\hat{L}_1^{\text{direct}} = \frac{1}{T^2}\left\langle \mathbf{K}_1\mathbf{K}_2 \right\rangle, \tag{416}$$

$$\hat{L}_2^{\text{direct}} = \frac{1}{T^4}\left(\mathbf{1}_T^*\mathbf{K}_1\mathbf{1}\right)\left(\mathbf{1}_T^*\mathbf{K}_2\mathbf{1}\right), \tag{417}$$

$$\hat{L}_3^{\text{direct}} = \frac{1}{T^3}\mathbf{1}_T^*\mathbf{K}_1\mathbf{K}_2\mathbf{1}_T, \tag{418}$$

$$\mathbf{K}_m \approx \mathbf{G}_m\mathbf{G}_m^*, \tag{419}$$

$$\hat{L}_1^{\text{iChol}} = \frac{1}{T^2}\mathbf{1}_{d_1}^*\left(\mathbf{G}_1^*\mathbf{G}_2 \circ \mathbf{G}_1^*\mathbf{G}_2\right)\mathbf{1}_{d_2}, \tag{420}$$

$$\hat{L}_2^{\text{iChol}} = \frac{1}{T^4}\left\|\mathbf{1}_T^*\mathbf{G}_1\right\|_2^2\left\|\mathbf{1}_T^*\mathbf{G}_2\right\|_2^2, \tag{421}$$

$$\hat{L}_3^{\text{iChol}} = \frac{1}{T^3}\left(\mathbf{1}_T^*\mathbf{G}_1\right)\left(\mathbf{G}_1^*\mathbf{G}_2\right)\left(\mathbf{G}_2^*\mathbf{1}_T\right). \tag{422}$$

– `QMI_CS_KDE_direct`:

$$k_m(\mathbf{u},\mathbf{v}) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}} \quad (\forall m), \tag{423}$$

$$\hat{I}_{\text{QMI-CS}}\left(\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right) = \log\left[\frac{\hat{L}_1^{\text{direct}}\hat{L}_2^{\text{direct}}}{(\hat{L}_3^{\text{direct}})^2}\right]. \tag{424}$$

(425)

– `QMI_CS_KDE_iChol`:

$$k_m(\mathbf{u},\mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}} \quad (\forall m), \tag{426}$$

$$\hat{I}_{\text{QMI-CS}}\left(\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right) = \log\left[\frac{\hat{L}_1^{\text{iChol}}\hat{L}_2^{\text{iChol}}}{(\hat{L}_3^{\text{iChol}})^2}\right]. \tag{427}$$

(428)

– `QMI_ED_KDE_iChol`:

$$k_m(\mathbf{u},\mathbf{v}) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^d}e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}} \quad (\forall m), \tag{429}$$

$$\hat{I}_{\text{QMI-ED}}\left(\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right) = \hat{L}_1^{\text{iChol}} + \hat{L}_2^{\text{iChol}} - 2\hat{L}_3^{\text{iChol}}. \tag{430}$$

- `dCov`, `dCor` [175, 174]: The estimation can be carried out on the basis of the pairwise distances of the sample points:

$$a_{kl} = \left\|\mathbf{y}_k^1 - \mathbf{y}_l^1\right\|_2^\alpha, \quad \bar{a}_{k\cdot} = \frac{1}{T}\sum_{l=1}^T a_{kl}, \quad \bar{a}_{\cdot l} = \frac{1}{T}\sum_{k=1}^T a_{kl}, \quad \bar{a}_{\cdot\cdot} = \frac{1}{T^2}\sum_{k,l=1}^T a_{kl}, \quad A_{kl} = a_{kl} - \bar{a}_{k\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}, \tag{431}$$

$$b_{kl} = \left\|\mathbf{y}_k^2 - \mathbf{y}_l^2\right\|_2^\alpha, \quad \bar{b}_{k\cdot} = \frac{1}{T}\sum_{l=1}^T b_{kl}, \quad \bar{b}_{\cdot l} = \frac{1}{T}\sum_{k=1}^T b_{kl}, \quad \bar{b}_{\cdot\cdot} = \frac{1}{T^2}\sum_{k,l=1}^T b_{kl}, \quad B_{kl} = b_{kl} - \bar{b}_{k\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}, \tag{432}$$

$$\hat{I}_{\text{dCov}}\left(\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right) = \frac{1}{T}\sqrt{\sum_{k,l=1}^T A_{kl}B_{kl}} = \frac{1}{T}\sqrt{\langle \mathbf{A}, \mathbf{B}\rangle}, \tag{433}$$

$$\hat{I}_{\text{dVar}}\left(\mathbf{Y}_{1:T}^1\right) = \frac{1}{T}\sqrt{\sum_{k,l=1}^T (A_{kl})^2} = \frac{1}{T}\sqrt{\langle \mathbf{A}, \mathbf{A}\rangle}, \tag{434}$$

$$\hat{I}_{\text{dVar}}\left(\mathbf{Y}_{1:T}^2\right) = \frac{1}{T}\sqrt{\sum_{k,l=1}^T (B_{kl})^2} = \frac{1}{T}\sqrt{\langle \mathbf{B}, \mathbf{B}\rangle}, \tag{435}$$

$$\hat{I}_{\mathrm{dCor}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T}\right)=\begin{cases}\frac{I_{\mathrm{dCov}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T}\right)}{\sqrt{I_{\mathrm{dVar}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^1_{1:T}\right)I_{\mathrm{dVar}}\left(\mathbf{Y}^2_{1:T},\mathbf{Y}^2_{1:T}\right)}}, & \text{if } \hat{I}_{\mathrm{dVar}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^1_{1:T}\right)I_{\mathrm{dVar}}\left(\mathbf{Y}^2_{1:T},\mathbf{Y}^2_{1:T}\right)>0,\\ 0, & \text{otherwise.}\end{cases} \tag{436}$$

- `3way_Lancaster` [139, 79]:

$$\mathbf{H}=[H_{ij}]=\left[\delta_{ij}-\frac{1}{T}\right]\in\mathbb{R}^{T\times T}, \tag{437}$$

$$\mathbf{K}_m=[(\mathbf{K}_m)_{ij}]=\left[k_m\left(\mathbf{y}^m_i,\mathbf{y}^m_j\right)\right],\quad (m=1,2,3), \tag{438}$$

$$\bar{\mathbf{K}}_m=\mathbf{H}\mathbf{K}_m\mathbf{H}, \tag{439}$$

$$\hat{I}_{3\text{-Lanc}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T},\mathbf{Y}^3_{1:T}\right)=\frac{\mathbf{1}^*_T\left(\bar{\mathbf{K}}_1\circ\bar{\mathbf{K}}_2\circ\bar{\mathbf{K}}_3\right)\mathbf{1}}{T^2}. \tag{440}$$

  Currently, k_m-s are RBF-s; the bandwiths can be chosen by the median heuristic.

- `3way_joint` [139]:

$$\mathbf{H}=[H_{ij}]=\left[\delta_{ij}-\frac{1}{T}\right]\in\mathbb{R}^{T\times T}, \tag{441}$$

$$\mathbf{K}_m=[(\mathbf{K}_m)_{ij}]=\left[k_m\left(\mathbf{y}^m_i,\mathbf{y}^m_j\right)\right],\quad (m=1,2,3), \tag{442}$$

$$(\mathbf{A})_+=\mathbf{1}_T\mathbf{1}^*_T\mathbf{A}\in\mathbb{R}^{T\times T},\quad (\mathbf{A}\in\mathbb{R}^{T\times T}) \tag{443}$$

$$(\mathbf{A})_{++}=\mathbf{1}^*_T\mathbf{A}\mathbf{1}_T=\sum_{i,j=1}^{T}A_{ij}\in\mathbb{R},\quad (\mathbf{A}\in\mathbb{R}^{T\times T}) \tag{444}$$

$$\hat{I}_{3\text{-joint}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T},\mathbf{Y}^3_{1:T}\right)=\frac{(\mathbf{K}_1\circ\mathbf{K}_2\circ\mathbf{K}_3)_{++}}{T^2}-\frac{2}{T^4}tr\left[(\mathbf{K}_1)_+\circ(\mathbf{K}_2)_+\circ(\mathbf{K}_3)_+\right]+\prod_{u=1}^{3}\frac{(\mathbf{K}_u)_{++}}{T^2}. \tag{445}$$

  Currently, k_m-s are RBF-s; the bandwiths can be chosen by the median heuristic.

- `Shannon_vME` [72]: In fact this is a 'hidden' meta estimator using Eq. (1) and Eqs. (390) - (391).

E.3 Divergence

Notations. 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. (71), (73)]:

$$D_{\mathrm{temp1}}(\alpha)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^{\alpha}\left[f_2(\mathbf{u})\right]^{1-\alpha}\mathrm{d}\mathbf{u}, \tag{446}$$

$$D_{\mathrm{temp2}}(a,b)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^{a}\left[f_2(\mathbf{u})\right]^{b}f_1(\mathbf{y})\mathrm{d}\mathbf{u}. \tag{447}$$

The definition of D_{temp3} is as follows:

$$D_{\mathrm{temp3}}(\alpha)=\int_{\mathbb{R}^d}f_1(\mathbf{u})f_2^{\alpha-1}(\mathbf{u})\mathrm{d}\mathbf{u}. \tag{448}$$

- `L2_kNN_k` [122, 120, 124]:

$$\hat{D}_{\mathrm{L}}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right)=\sqrt{\frac{1}{T_1V_d}\sum_{t=1}^{T_1}\left[\frac{k-1}{(T_1-1)\rho^d_k(t)}-\frac{2(k-1)}{T_2\nu^d_k(t)}+\frac{(T_1-1)\rho^d_k(t)(k-2)(k-1)}{(T_2)^2\nu^{2d}_k(t)k}\right]}. \tag{449}$$

- `Tsallis_kNN_k` [122, 120]:

$$B_{k,\alpha} = \frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}, \tag{450}$$

$$\hat{D}_{\text{temp1}}\left(\alpha; \mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = B_{k,\alpha}\frac{(T_1-1)^{1-\alpha}}{(T_2)^{1-\alpha}}\frac{1}{T_1}\sum_{t=1}^{T_1}\left[\frac{\rho_k(t)}{\nu_k(t)}\right]^{d(1-\alpha)}, \tag{451}$$

$$\hat{D}_{\text{T},\alpha}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \frac{1}{\alpha-1}\left[\hat{D}_{\text{temp1}}\left(\alpha; \mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) - 1\right]. \tag{452}$$

- `Renyi_kNN_k` [122, 120, 124]:

$$B_{k,\alpha} = \frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}, \tag{453}$$

$$\hat{D}_{\text{temp1}}\left(\alpha; \mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = B_{k,\alpha}\frac{(T_1-1)^{1-\alpha}}{(T_2)^{1-\alpha}}\frac{1}{T_1}\sum_{t=1}^{T_1}\left[\frac{\rho_k(t)}{\nu_k(t)}\right]^{d(1-\alpha)}, \tag{454}$$

$$\hat{D}_{\text{R},\alpha}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \frac{1}{\alpha-1}\log\left[\hat{D}_{\text{temp1}}\left(\alpha; \mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right)\right]. \tag{455}$$

- `MMD_Ustat` [49]:

$$t_1 = \frac{1}{T_1(T_1-1)}\sum_{i=1}^{T_1}\sum_{j=1;j\neq i}^{T_1} k\left(\mathbf{y}^1_i, \mathbf{y}^1_j\right), \tag{456}$$

$$t_2 = \frac{1}{T_2(T_2-1)}\sum_{i=1}^{T_2}\sum_{j=1;j\neq i}^{T_2} k\left(\mathbf{y}^2_i, \mathbf{y}^2_j\right), \tag{457}$$

$$t_3 = \frac{2}{T_1T_2}\sum_{i=1}^{T_1}\sum_{j=1}^{T_2} k\left(\mathbf{y}^1_i, \mathbf{y}^2_j\right), \tag{458}$$

$$\hat{D}_{\text{MMD}}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \sqrt{t_1+t_2-t_3}. \tag{459}$$

  The estimator supports the (271) - (275) k kernels.

- `MMD_Vstat` [49]:

$$k(\mathbf{u}, \mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}}, \tag{460}$$

$$t_1 = \frac{1}{(T_1)^2}\sum_{i,j=1}^{T_1} k\left(\mathbf{y}^1_i, \mathbf{y}^1_j\right), \tag{461}$$

$$t_2 = \frac{1}{(T_2)^2}\sum_{i,j=1}^{T_2} k\left(\mathbf{y}^2_i, \mathbf{y}^2_j\right), \tag{462}$$

$$t_3 = \frac{2}{T_1T_2}\sum_{i=1}^{T_1}\sum_{j=1}^{T_2} k\left(\mathbf{y}^1_i, \mathbf{y}^2_j\right), \tag{463}$$

$$\hat{D}_{\text{MMD}}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \sqrt{t_1+t_2-t_3}. \tag{464}$$

- `MMD_Ustat_iChol` [49]:

$$k(\mathbf{u}, \mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}}, \tag{465}$$

$$[\mathbf{y}_1, \ldots, \mathbf{y}_T] = \left[\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right], \quad (T = T_1+T_2), \tag{466}$$

$$\mathbf{K} = \left[k\left(\mathbf{y}_i, \mathbf{y}_j\right)\right] \approx \mathbf{L}\mathbf{L}^* =: \left[\mathbf{L}_1; \mathbf{L}_2\right]\left[\mathbf{L}_1; \mathbf{L}_2\right]^*, \quad \left(\mathbf{L} \in \mathbb{R}^{T\times d_c}, \mathbf{L}_1 \in \mathbb{R}^{T_1\times d_c}, \mathbf{L}_2 \in \mathbb{R}^{T_2\times d_c}\right) \tag{467}$$

$$\mathbf{l}_m = \mathbf{1}_{T_m}^* \mathbf{L}_m \in \mathbb{R}^{1\times d_c}, \quad (m=1,2), \tag{468}$$

$$t_m = \frac{\mathbf{l}_m \mathbf{l}_m^* - \sum_{i=1}^{T_m}\sum_{j=1}^{d}\left[(\mathbf{L}_m)_{ij}\right]^2}{T_m(T_m-1)}, \quad (m=1,2), \tag{469}$$

$$t_3 = \frac{2\mathbf{l}_1\mathbf{l}_2^*}{T_1 T_2}, \tag{470}$$

$$\hat{D}_{\mathrm{MMD}}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = \sqrt{t_1 + t_2 - t_3}. \tag{471}$$

- `MMD_Vstat_iChol` [49]:

$$k(\mathbf{u},\mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}}, \tag{472}$$

$$[\mathbf{y}_1,\ldots,\mathbf{y}_T] = \left[\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right], \quad (T = T_1+T_2), \tag{473}$$

$$\mathbf{K} = \left[k\left(\mathbf{y}_i,\mathbf{y}_j\right)\right] \approx \mathbf{L}\mathbf{L}^* =: \left[\mathbf{L}_1;\mathbf{L}_2\right]\left[\mathbf{L}_1;\mathbf{L}_2\right]^*, \quad \left(\mathbf{L}\in\mathbb{R}^{T\times d_c}, \mathbf{L}_1\in\mathbb{R}^{T_1\times d_c}, \mathbf{L}_2\in\mathbb{R}^{T_2\times d_c}\right) \tag{474}$$

$$\mathbf{l}_m = \mathbf{1}_{T_m}^* \mathbf{L}_m \in \mathbb{R}^{1\times d_c}, \quad (m=1,2), \tag{475}$$

$$t_m = \frac{\mathbf{l}_m \mathbf{l}_m^*}{(T_m)^2}, \quad (m=1,2), \tag{476}$$

$$t_3 = \frac{2\mathbf{l}_1\mathbf{l}_2^*}{T_1 T_2}, \tag{477}$$

$$\hat{D}_{\mathrm{MMD}}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = \sqrt{t_1 + t_2 - t_3}. \tag{478}$$

- `MMD_online` [49]:

$$T' = \left\lfloor \frac{T_1}{2} \right\rfloor \left(= \left\lfloor \frac{T_2}{2} \right\rfloor\right), \tag{479}$$

$$h((\mathbf{x},\mathbf{y}),(\mathbf{u},\mathbf{v})) = k(\mathbf{x},\mathbf{u}) + k(\mathbf{y},\mathbf{v}) - k(\mathbf{x},\mathbf{v}) - k(\mathbf{y},\mathbf{u}), \tag{480}$$

$$\hat{D}_{\mathrm{MMD}}\left(\mathbf{Y}_{1:T}^1, \mathbf{Y}_{1:T}^2\right) = \frac{1}{T'}\sum_{t=1}^{T'} h\left(\left(\mathbf{y}_{2t-1}^1, \mathbf{y}_{2t-1}^2\right), \left(\mathbf{y}_{2t}^1, \mathbf{y}_{2t}^2\right)\right). \tag{481}$$

  Currently, k is RBF.

- `Hellinger_kNN_k` [125]:

$$B_{k,a,b} = V_d^{-(a+b)} \frac{\Gamma(k)^2}{\Gamma(k-a)\Gamma(k-b)}, \tag{482}$$

$$\hat{D}_{\mathrm{temp2}}\left(a,b;\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = (T_1-1)^{-a}(T_2)^{-b} B_{k,a,b} \frac{1}{T_1}\sum_{t=1}^{T_1}[\rho_k(t)]^{-da}[\nu_k(t)]^{-db}, \tag{483}$$

$$\hat{D}_{\mathrm{H}}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = \sqrt{1 - \hat{D}_{\mathrm{temp2}}\left(-\frac{1}{2},\frac{1}{2};\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right)}. \tag{484}$$

- `Bhattacharyya_kNN_k` [12, 125]:

$$B_{k,a,b} = V_d^{-(a+b)} \frac{\Gamma(k)^2}{\Gamma(k-a)\Gamma(k-b)}, \tag{485}$$

$$\hat{D}_{\mathrm{temp2}}\left(a,b;\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = (T_1-1)^{-a}(T_2)^{-b} B_{k,a,b} \frac{1}{T_1}\sum_{t=1}^{T_1}[\rho_k(t)]^{-da}[\nu_k(t)]^{-db}, \tag{486}$$

$$\hat{D}_{\mathrm{B}}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = -\log\left[\hat{D}_{\mathrm{temp2}}\left(-\frac{1}{2},\frac{1}{2};\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right)\right]. \tag{487}$$

- `KL_kNN_k` [81, 114, 190]:

$$\hat{D}\left(\mathbf{Y}_{1:T_1}^1, \mathbf{Y}_{1:T_2}^2\right) = \frac{d}{T_1}\sum_{t=1}^{T_1}\log\left[\frac{\nu_k(t)}{\rho_k(t)}\right] + \log\left(\frac{T_2}{T_1-1}\right). \tag{488}$$

- `KL_kNN_kiTi` [190]:

$$k_1 = k_1(T_1) = \left\lfloor \sqrt{T_1} \right\rfloor, \tag{489}$$

$$k_2 = k_2(T_2) = \left\lfloor \sqrt{T_2} \right\rfloor, \tag{490}$$

$$\hat{D}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \frac{1}{T_1}\sum_{t=1}^{T_1} \log\left[\frac{k_1}{k_2}\frac{T_2}{T_1-1}\frac{\nu^d_{k_2}(t)}{\rho^d_{k_1}(t)}\right] = \frac{d}{T_1}\sum_{t=1}^{T_1}\log\left[\frac{\nu_{k_2}(t)}{\rho_{k_1}(t)}\right] + \log\left(\frac{k_1}{k_2}\frac{T_2}{T_1-1}\right). \tag{491}$$

- `CS_KDE_iChol`, `ED_KDE_iChol` [142];

$$\mathbf{Z}_{1:2T} = \left[\mathbf{Y}^1_{1:T}, \mathbf{Y}^2_{1:T}\right], \tag{492}$$

$$k(\mathbf{u}, \mathbf{v}) = \frac{1}{\left(\sqrt{2\pi}\sigma\right)^d} e^{\frac{-\|\mathbf{u}-\mathbf{v}\|_2^2}{2\sigma^2}}, \tag{493}$$

$$(\mathbf{K})_{ij} = k(\mathbf{z}_i, \mathbf{z}_j), \tag{494}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \in \mathbb{R}^{(2T)\times(2T)}, \tag{495}$$

$$\mathbf{K} \approx \mathbf{G}\mathbf{G}^*, \tag{496}$$

$$D_{\mathrm{CS}}(f_1, f_2) = \log\left[\frac{L_1 L_2}{(L_3)^2}\right], \tag{497}$$

$$D_{\mathrm{ED}}(f_1, f_2) = L_1 + L_2 - 2L_3, \tag{498}$$

$$\mathbf{e}_1 = [\mathbf{1}_T; \mathbf{0}_T], \tag{499}$$

$$\mathbf{e}_2 = [\mathbf{0}_T; \mathbf{1}_T], \tag{500}$$

$$\hat{L}_1 = \frac{1}{T^2}(\mathbf{e}_1^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_1), \tag{501}$$

$$\hat{L}_2 = \frac{1}{T^2}(\mathbf{e}_2^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_2), \tag{502}$$

$$\hat{L}_3 = \frac{1}{T^2}(\mathbf{e}_1^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_2), \tag{503}$$

$$\hat{D}_{\mathrm{CS}}\left(\mathbf{Y}^1_{1:T}, \mathbf{Y}^2_{1:T}\right) = \log\left[\frac{\hat{L}_1\hat{L}_2}{(\hat{L}_3)^2}\right], \tag{504}$$

$$\hat{D}_{\mathrm{ED}}\left(\mathbf{Y}^1_{1:T}, \mathbf{Y}^2_{1:T}\right) = \hat{L}_1 + \hat{L}_2 - 2\hat{L}_3. \tag{505}$$

- `EnergyDist` [172, 173]:

$$\hat{D}_{\mathrm{EnDist}}(f_1, f_2) = \frac{2}{T_1T_2}\sum_{t_1=1}^{T_1}\sum_{t_2=1}^{T_2}\rho\left(\mathbf{y}^1_{t_1}, \mathbf{y}^2_{t_2}\right) - \frac{1}{(T_1)^2}\sum_{t_1=1}^{T_1}\sum_{t_2=1}^{T_1}\rho\left(\mathbf{y}^1_{t_1}, \mathbf{y}^1_{t_2}\right) - \frac{1}{(T_2)^2}\sum_{t_1=1}^{T_2}\sum_{t_2=1}^{T_2}\rho\left(\mathbf{y}^2_{t_1}, \mathbf{y}^2_{t_2}\right). \tag{506}$$

- `Bregman_kNN_k` [15, 27, 81]:

$$\hat{D}_{\mathrm{temp3}}\left(\alpha; \mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \frac{1}{T_1}\sum_{t=1}^{T_1}\left[T_2 C_{\alpha,k} V_d \nu^d_k(t)\right]^{1-\alpha} = \frac{T_2^{1-\alpha} C^{1-\alpha}_{\alpha,k} V_d^{1-\alpha}}{T_1}\sum_{t=1}^{T_1}\nu_k^{d(1-\alpha)}(t), \tag{507}$$

$$\hat{D}_{\mathrm{NB},\alpha}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \hat{I}_\alpha\left(\mathbf{Y}^2_{1:T_2}\right) + \frac{1}{\alpha-1}\hat{I}_\alpha\left(\mathbf{Y}^1_{1:T_1}\right) - \frac{\alpha}{\alpha-1}\hat{D}_{\mathrm{temp3}}\left(\alpha; \mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right). \tag{508}$$

  where the I_α and the D_{temp3} quantities are defined in Eq. (277) and Eq. (448).

- `symBregman_kNN_k` [15, 27, 81], via Eq. (66):

$$\hat{D}_{\mathrm{SB},\alpha}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \frac{1}{\alpha-1}\left[\hat{I}_\alpha\left(\mathbf{Y}^1_{1:T_1}\right) + \hat{I}_\alpha\left(\mathbf{Y}^2_{1:T_2}\right) - \hat{D}_{\mathrm{temp3}}\left(\alpha; \mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) - \hat{D}_{\mathrm{temp3}}\left(\alpha; \mathbf{Y}^2_{1:T_2}, \mathbf{Y}^1_{1:T_1}\right)\right], \tag{509}$$

  where the I_α and the D_{temp3} quantities are defined in Eq. (277) and Eq. (448).

- `ChiSquare_kNN_k` [125]:

$$B_{k,a,b} = V_d^{-(a+b)} \frac{\Gamma(k)^2}{\Gamma(k-a)\Gamma(k-b)}, \tag{510}$$

$$\hat{D}_{\text{temp2}}\left(a,b;\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = (T_1-1)^{-a}(T_2)^{-b}B_{k,a,b}\frac{1}{T_1}\sum_{t=1}^{T_1}[\rho_k(t)]^{-da}[\nu_k(t)]^{-db}, \tag{511}$$

$$\hat{D}_{\chi^2}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = \hat{D}_{\text{temp2}}\left(1,-1;\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) - 1. \tag{512}$$

- `KL_PSD_SzegoT` [47, 46, 127]: Similarly to `Shannon_PSD_SzegoT`, the method assumes that $supp(f_m) \subseteq \left[-\frac{1}{2},\frac{1}{2}\right]$ $(m=1,2)$ [this is assured for the samples by the $y_t^m \to \frac{y_t}{2\max(a_1,a_2)}$ pre-processing, where $a_m = \max_{t=1,\ldots,T}|y_t^m|$ $(m=1,2)$]. Let

$$\mathbf{\Phi}_K^{(f_m)} = \left[\int_{-\frac{1}{2}}^{\frac{1}{2}} f_m(u)e^{i2\pi u(a-b)}\mathrm{d}u\right]_{a,b=1}^{K} = \left[\mathbb{E}\left(e^{i2\pi y^m(a-b)}\right)\right]_{a,b=1}^{K}, \quad (m=1,2) \tag{513}$$

and their empirical estimations be

$$\hat{\mathbf{\Phi}}_K^{(f_m)} = \left[\frac{1}{T_m}\sum_{t=1}^{T} e^{i2\pi y_t^m(a-b)}\right]_{a,b=1}^{K}, \quad (m=1,2). \tag{514}$$

Let $\lambda_{K,k}$ and $\beta_{K,k}$ denote the eigenvalues of $\mathbf{\Phi}_K^{(f_1)}$ and $\mathbf{\Phi}_K^{(f_1)}\log\left[\mathbf{\Phi}_K^{(f_2)}\right]$, respectively. The estimator implements the approximation

$$\hat{D}\left(Y^1_{1:T_1},Y^2_{1:T_2}\right) = \frac{1}{K}\sum_{k=1}^{K}\lambda_{K,k}\log\left(\lambda_{K,k}\right) - \frac{1}{K}\sum_{k=1}^{K}\beta_{K,k}. \tag{515}$$

- `BMMD_DMMD_Ustat` [197]: This measure is defined as the average of U-statistic based MMD estimators over B-sized blocks:

$$\hat{D}_{\text{MMD}}\left(\mathbf{Y}^1_{1:T},\mathbf{Y}^2_{1:T}\right) = \frac{1}{\left\lfloor\frac{T}{B}\right\rfloor}\sum_{k=1}^{\left\lfloor\frac{T}{B}\right\rfloor}\hat{D}_{\text{MMD}}\left(\mathbf{Y}^1_{(k-1)B+1:kB},\mathbf{Y}^2_{(k-1)B+1:kB}\right). \tag{516}$$

It combines the favourable properties of linear and U-statistic based estimators (powerful, computationally efficient).

- `Sharma_expF` [103]: MLE estimation $(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2)$ plugged into (246).

- `Sharma_kNN_k`: [122, 120]:

$$B_{k,\alpha} = \frac{\Gamma(k)^2}{\Gamma(k-\alpha+1)\Gamma(k+\alpha-1)}, \tag{517}$$

$$\hat{D}_{\text{temp1}}\left(\alpha;\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = B_{k,\alpha}\frac{(T_1-1)^{1-\alpha}}{(T_2)^{1-\alpha}}\frac{1}{T_1}\sum_{t=1}^{T_1}\left[\frac{\rho_k(t)}{\nu_k(t)}\right]^{d(1-\alpha)}, \tag{518}$$

$$\hat{D}_{\text{SM},\alpha,\beta}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = \frac{1}{\beta-1}\left(\left[\hat{D}_{\text{temp1}}\left(\alpha;\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right)\right]^{\frac{1-\beta}{1-\alpha}} - 1\right). \tag{519}$$

- `KL_expF` [101]: MLE estimation $(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2)$ plugged into (251).

- `ChiSquare_expF` [104]: MLE estimation $(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2)$ plugged into (253).

- `KL_vME`, `Renyi_vME`, `Tsallis_vME`, `ChiSquare_vME`, `Hellinger_vME` [72]: Let T be the divergence of interest and (ψ_1,ψ_2) be the influence function associated to T,

$$T\left(f^1,f^2\right) = T\left(\widehat{f^1},\widehat{f^2}\right) + \frac{1}{T_1}\sum_{t=1}^{T_1}\psi_1\left(\mathbf{y}^1_t;\widehat{f^1},\widehat{f^2}\right) + \frac{1}{T_2}\sum_{t=1}^{T_2}\psi_2\left(\mathbf{y}^2_t;\widehat{f^1},\widehat{f^2}\right). \tag{520}$$

E.4 Association Measures

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

- `Spearman1`, `Spearman2`, `Spearman3` [150, 194, 96, 135, 67, 98, 97, 136]: One can arrive at explicit formulas by substituting the empirical copula of $\mathbf{y}$ ($\hat{C}_T$, see Eq. (396)) to the definitions of $\hat{A}_{\rho_i}$-s ($i = 1, 2, 3$; see Eqs. (79), (81), (82)). The resulting nonparametric estimators are

$$\hat{A}_{\rho_1}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_1}(\hat{C}_T) = h_\rho(d)\left[2^d \int_{[0,1]^d} \hat{C}_T(\mathbf{u})\mathrm{d}\mathbf{u} - 1\right] = h_\rho(d)\left[\frac{2^d}{T}\sum_{j=1}^{T}\prod_{i=1}^{d}(1-\hat{U}_{ij}) - 1\right], \tag{521}$$

$$\hat{A}_{\rho_2}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_2}(\hat{C}_T) = h_\rho(d)\left[2^d \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\hat{C}_T(\mathbf{u}) - 1\right] = h_\rho(d)\left[\frac{2^d}{T}\sum_{j=1}^{T}\prod_{i=1}^{d}\hat{U}_{ij} - 1\right], \tag{522}$$

$$\hat{A}_{\rho_3}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_3}(\hat{C}_T) = \frac{\hat{A}_{\rho_1}(\mathbf{Y}_{1:T}) + \hat{A}_{\rho_2}(\mathbf{Y}_{1:T})}{2}, \tag{523}$$

  where $h_\rho(d)$ and $\hat{U}_{ij}$ are defined in Eq. (80) and Eq. (394), respectively.

- `Spearman4` [73, 135]:

$$\hat{A}_{\rho_4}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_4}(\hat{C}_T) = \frac{12}{T}\binom{d}{2}^{-1}\sum_{k,l=1;k<l}^{d}\sum_{j=1}^{T}(1-\hat{U}_{kj})(1-\hat{U}_{lj}) - 3, \tag{524}$$

  where $\hat{U}_{kj}$ and $\hat{U}_{lj}$ are defined in Eq. (394).

- `CorrEntr_KDE_direct` [128]:

$$k(u,v) = e^{-\frac{(u-v)^2}{2\sigma^2}}, \tag{525}$$

$$\mathbf{Y}_{1:T} = \left[\mathbf{y}^1_{1:T}; \mathbf{y}^2_{1:T}\right], \quad (\mathbf{y}^i_{1:T} \in \mathbb{R}^{1\times T}) \tag{526}$$

$$\hat{A}_{\mathrm{CorrEntr}}\left(\mathbf{Y}_{1:T}\right) = \frac{1}{T}\sum_{t=1}^{T} k\left(y^1_t, y^2_t\right). \tag{527}$$

- `CCorrEntr_KDE_iChol` [128, 142]:

$$k(u,v) = e^{-\frac{(u-v)^2}{2\sigma^2}}, \tag{528}$$

$$\mathbf{Y}_{1:T} = \left[\mathbf{y}^1_{1:T}; \mathbf{y}^2_{1:T}\right], \quad (\mathbf{y}^i_{1:T} \in \mathbb{R}^{1\times T}) \tag{529}$$

$$\mathbf{Z}_{1:2T} = \left[\mathbf{y}^1_{1:T}, \mathbf{y}^2_{1:T}\right], \tag{530}$$

$$(\mathbf{K})_{ij} = k(\mathbf{z}_i, \mathbf{z}_j), \tag{531}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \in \mathbb{R}^{(2T)\times(2T)}, \tag{532}$$

$$\mathbf{K} \approx \mathbf{G}\mathbf{G}^*, \tag{533}$$

$$\mathbf{e}_1 = [\mathbf{1}_T; \mathbf{0}_T], \tag{534}$$

$$\mathbf{e}_2 = [\mathbf{0}_T; \mathbf{1}_T], \tag{535}$$

$$L = \sum_{t_1=1}^{T}\sum_{t_2=1}^{T} k\left(y^1_{t_1}, y^2_{t_2}\right), \tag{536}$$

$$\hat{L} = \frac{1}{T^2}(\mathbf{e}_1^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_2), \tag{537}$$

$$\hat{A}_{\mathrm{CCorrEntr}}\left(\mathbf{Y}_{1:T}\right) = \frac{1}{T}\sum_{t=1}^{T} k\left(y^1_t, y^2_t\right) - \frac{\hat{L}}{T^2}. \tag{538}$$

- `CCorrEntr_KDE_Lapl` [128, 21]:

$$k(u,v) = e^{-\frac{|u-v|}{\sigma}}, \tag{539}$$

$$\mathbf{Y}_{1:T} = \left[\mathbf{y}^1_{1:T}; \mathbf{y}^2_{1:T}\right], \quad (\mathbf{y}^i_{1:T} \in \mathbb{R}^{1\times T}) \tag{540}$$

$$L = \sum_{t_1=1}^{T}\sum_{t_2=1}^{T} k\left(y^1_{t_1}, y^2_{t_2}\right) = \sum_{t_1=1}^{T}\sum_{t_2=1}^{T} e^{-\frac{\left|y^1_{t_1} - y^2_{t_2}\right|}{\sigma}} \tag{541}$$

$$= \sum_{t_1=1}^{T}\left[e^{-\frac{y^1_{t_1}}{\sigma}} \sum_{\{t_2: y^2_{t_2} \le y^1_{t_1}\}} e^{\frac{y^2_{t_2}}{\sigma}} + e^{\frac{y^1_{t_1}}{\sigma}} \sum_{\{t_2: y^2_{t_2} > y^1_{t_1}\}} e^{-\frac{y^2_{t_2}}{\sigma}}\right] = [21], \tag{542}$$

$$\hat{A}_{\text{CCorrEntr}}\left(\mathbf{Y}_{1:T}\right) = \frac{1}{T}\sum_{t=1}^{T} k\left(y^1_t, y^2_t\right) - \frac{L}{T^2}. \tag{543}$$

- `CorrEntrCoeff_KDE_direct` [128]:

$$k(u,v) = e^{-\frac{(u-v)^2}{2\sigma^2}}, \tag{544}$$

$$\mathbf{Y}_{1:T} = \left[\mathbf{y}^1_{1:T}; \mathbf{y}^2_{1:T}\right], \quad (\mathbf{y}^i_{1:T} \in \mathbb{R}^{1\times T}) \tag{545}$$

$$C = \frac{1}{T}\sum_{t=1}^{T} k\left(y^1_t, y^2_t\right), \tag{546}$$

$$\hat{A}_{\text{CorrEntrCoeff}}\left(\mathbf{Y}_{1:T}\right) = \frac{C - \frac{\mathbf{1}^*_T \mathbf{K}_{12} \mathbf{1}_T}{T^2}}{\sqrt{\left(1 - \frac{\mathbf{1}^*_T K_{11} \mathbf{1}_T}{T^2}\right)\left(1 - \frac{\mathbf{1}^*_T K_{22} \mathbf{1}_T}{T^2}\right)}}. \tag{547}$$

- `CorrEntrCoeff_KDE_iChol` [128, 142]:

$$\mathbf{Y}_{1:T} = \left[\mathbf{y}^1_{1:T}; \mathbf{y}^2_{1:T}\right], \quad (\mathbf{y}^i_{1:T} \in \mathbb{R}^{1\times T}) \tag{548}$$

$$\mathbf{Z}_{1:2T} = \left[\mathbf{y}^1_{1:T}, \mathbf{y}^2_{1:T}\right], \tag{549}$$

$$k(u,v) = e^{-\frac{(u-v)^2}{2\sigma^2}}, \tag{550}$$

$$(\mathbf{K})_{ij} = k(\mathbf{z}_i, \mathbf{z}_j), \tag{551}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \in \mathbb{R}^{(2T)\times(2T)}, \tag{552}$$

$$\mathbf{K} \approx \mathbf{G}\mathbf{G}^*, \tag{553}$$

$$C = \frac{1}{T}\sum_{t=1}^{T} k\left(y^1_t, y^2_t\right), \tag{554}$$

$$\mathbf{e}_1 = [\mathbf{1}_T; \mathbf{0}_T]/T, \tag{555}$$

$$\mathbf{e}_2 = [\mathbf{0}_T; \mathbf{1}_T]/T, \tag{556}$$

$$\hat{A}_{\text{CorrEntrCoeff}}\left(\mathbf{Y}_{1:T}\right) = \frac{C - (\mathbf{e}_2^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_1)}{\sqrt{\left[1 - (\mathbf{e}_1^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_1)\right]\left[1 - (\mathbf{e}_2^*\mathbf{G})(\mathbf{G}^*\mathbf{e}_2)\right]}}. \tag{557}$$

- `Blomqvist` [182, 136]: The empirical estimation of the survival function $\bar{C}$ is

$$\hat{\bar{C}}_T(\mathbf{u}) = \frac{1}{T}\sum_{t=1}^{T}\prod_{i=1}^{d} \mathbb{I}_{\{\hat{U}_{it} > u_i\}}, \quad (\mathbf{u} = [u_1; \ldots; u_d] \in [0,1]^d). \tag{558}$$

The estimation of Blomqvist's β is computed as

$$\mathbf{1/2} = \left[\frac{1}{2}; \ldots; \frac{1}{2}\right] \in \mathbb{R}^d, \tag{559}$$

$$h_\beta(d) = \frac{2^{d-1}}{2^{d-1}-1}, \tag{560}$$

$$A_\beta\left(y^1,\dots,y^d\right) = A_\beta(C) = h_\beta(d)\left[\hat{C}_T(\mathbf{1/2}) + \hat{\bar{C}}_T(\mathbf{1/2}) - 2^{1-d}\right]. \tag{561}$$

- `Spearman_lt` [134]:

$$\hat{A}_{\rho_{\text{lt}}}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_{\text{lt}}}(\hat{C}_T) = \frac{\frac{1}{T}\sum_{j=1}^{T}\prod_{i=1}^{d}(p-\hat{U}_{ij})^+ - \left(\frac{p^2}{2}\right)^d}{\frac{p^{d+1}}{d+1} - \left(\frac{p^2}{2}\right)^d}, \tag{562}$$

  where $\hat{U}_{ij}$ is defined in Eq. (394) and $z^+ = \max(z,0)$.

- `Spearman_L` [134]:

$$k = k(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{563}$$

$$\hat{A} = \hat{A}_{\rho_{\text{lt}}}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_{\text{lt}}}(\hat{C}_T) \text{ with } p = \frac{k}{T} \text{ in Eq. (562)}, \tag{564}$$

$$\hat{A}_{\rho_{\text{L}}}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_{\text{L}}}(\hat{C}_T) = \hat{A}. \tag{565}$$

- `Spearman_ut` [134]: For the estimation we need three quantities, that we provide below (they were not computed in [134]):

$$\int_{[1-p,1]^d} \hat{C}_T(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{1}{T}\sum_{j=1}^{T}\prod_{i=1}^{d}\left[1-\max(\hat{U}_{ij},1-p)\right] =: c, \tag{566}$$

$$\int_{[1-p,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u} = \left[\frac{p(2-p)}{2}\right]^d =: c_1(p,d), \tag{567}$$

$$\int_{[1-p,1]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{p^d(d+1-pd)}{d+1} =: c_2(p,d), \tag{568}$$

  where $\hat{U}_{ij}$ is defined in Eq. (394). Having these expressions at hand, the estimation can be simply written as [see Eq. (95)]:

$$\hat{A}_{\rho_{\text{ut}}}\left(\mathbf{Y}_{1:T}\right) = \hat{A}_{\rho_{\text{ut}}}(\hat{C}_T) = \frac{c - c_1(p,d)}{c_2(p,d) - c_1(p,d)}. \tag{569}$$

- `Spearman_U` [134]:

$$k = k(T) = \left\lfloor \sqrt{T} \right\rfloor, \tag{570}$$

$$\hat{A} = \hat{A}_{\rho_{\text{ut}}}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_{\text{ut}}}(\hat{C}_T) \text{ with } p = \frac{k}{T} \text{ in Eq. (569)}, \tag{571}$$

$$\hat{A}_{\rho_{\text{U}}}(\mathbf{Y}_{1:T}) = \hat{A}_{\rho_{\text{U}}}(\hat{C}_T) = \hat{A}. \tag{572}$$

E.5 Cross Quantities

Notations. We have T_1 and T_2 i.i.d. samples from the two distributions (f_1, f_2) to be compared: $\mathbf{Y}^1_{1:T_1} = \left(\mathbf{y}^1_1,\dots,\mathbf{y}^1_{T_1}\right)$, $\mathbf{Y}^2_{1:T_2} = \left(\mathbf{y}^2_1,\dots,\mathbf{y}^2_{T_2}\right)$ $(\mathbf{y}^i_t \in \mathbb{R}^d)$. 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` [81]:

$$\hat{C}_{\text{CE}}\left(\mathbf{Y}^1_{1:T_1}, \mathbf{Y}^2_{1:T_2}\right) = \log(V_d) + \log(T_2) - \psi(k) + \frac{d}{T_1}\sum_{t=1}^{T_1}\log\left[\nu_k(t)\right]. \tag{573}$$

- `CE_expF` [102]: MLE estimation $(\hat{\boldsymbol{\theta}})$ plugged into (245).

E.6 Kernels on Distributions

Notations. We have T_1 and T_2 i.i.d. samples from the two distributions $(f_1,\ f_2)$ whose similarity (kernel value) is to be estimated: $\mathbf{Y}^1_{1:T_1} = \left(\mathbf{y}^1_1,\dots,\mathbf{y}^1_{T_1}\right)$, $\mathbf{Y}^2_{1:T_2} = \left(\mathbf{y}^2_1,\dots,\mathbf{y}^2_{T_2}\right)$ $(\mathbf{y}^i_t \in \mathbb{R}^d)$. 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 $\mathbf{Y}^2_{1:T_2}\backslash\{\mathbf{y}^1_t\}$.

- '`expected`' [54, 43, 50, 94]:

$$\hat{K}_{\exp}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = \frac{1}{T_1T_2}\sum_{i=1}^{T_1}\sum_{j=1}^{T_2} k\left(\mathbf{y}^1_i,\mathbf{y}^2_j\right). \tag{574}$$

  The estimator supports the (271) - (275) k kernels.

- '`Bhattacharyya_kNN_k`' [12, 65, 125]:

$$B_{k,a,b} = V_d^{-(a+b)}\frac{\Gamma(k)^2}{\Gamma(k-a)\Gamma(k-b)}, \tag{575}$$

$$\hat{D}_{\text{temp2}}\left(a,b;\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = (T_1-1)^{-a}(T_2)^{-b}B_{k,a,b}\frac{1}{T_1}\sum_{t=1}^{T_1}[\rho_k(t)]^{-da}[\nu_k(t)]^{-db}, \tag{576}$$

$$\hat{K}_{\text{B}}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = \hat{D}_{\text{temp2}}\left(-\frac{1}{2},\frac{1}{2};\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right). \tag{577}$$

- '`PP_kNN_k`' [65, 125]:

$$B_{k,a,b} = V_d^{-(a+b)}\frac{\Gamma(k)^2}{\Gamma(k-a)\Gamma(k-b)}, \tag{578}$$

$$\hat{D}_{\text{temp2}}\left(a,b;\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = (T_1-1)^{-a}(T_2)^{-b}B_{k,a,b}\frac{1}{T_1}\sum_{t=1}^{T_1}[\rho_k(t)]^{-da}[\nu_k(t)]^{-db}, \tag{579}$$

$$\hat{K}_{\text{PP},\rho}\left(\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right) = \hat{D}_{\text{temp2}}\left(\rho-1,\rho;\mathbf{Y}^1_{1:T_1},\mathbf{Y}^2_{1:T_2}\right). \tag{580}$$

F Quick Tests for the Estimators: Derivations

In this section the derivations of relation (199), (201), (203), (204), (215) and (217) are detailed:

- Eq. (199), (201), (203): Let $\mathbf{y}\sim U[\mathbf{a},\mathbf{b}]$, then

$$H_{\text{R},\alpha}(\mathbf{y}) = \frac{1}{1-\alpha}\log\left[\int_{\mathbb{R}^d}\left(\frac{\mathbb{I}_{[\mathbf{a},\mathbf{b}]}(\mathbf{u})}{\prod_{i=1}^d(b_i-a_i)}\right)^{\alpha}\mathrm{d}\mathbf{u}\right] = \frac{1}{1-\alpha}\log\left(\int_{\mathbf{a}}^{\mathbf{b}}\frac{1}{\left[\prod_{i=1}^d(b_i-a_i)\right]^{\alpha}}\mathrm{d}\mathbf{u}\right) = \tag{581}$$

$$= \frac{1}{1-\alpha}\log\left(\left[\prod_{i=1}^d(b_i-a_i)\right]\frac{1}{\left[\prod_{i=1}^d(b_i-a_i)\right]^{\alpha}}\right) = \frac{1}{1-\alpha}\log\left(\left[\prod_{i=1}^d(b_i-a_i)\right]^{1-\alpha}\right) = \tag{582}$$

$$= \log\left[\prod_{i=1}^d(b_i-a_i)\right]. \tag{583}$$

  Using Eq. (4), relation (583) is also valid for $H(\mathbf{y})$, the Shannon entropy of $\mathbf{y}$. By the obtained formula for the Rényi entropy and Eq. (107) we get

$$H_{\text{T},\alpha}(\mathbf{y}) = \frac{e^{(1-\alpha)H_{\text{R},\alpha}(\mathbf{y})}-1}{1-\alpha} = \frac{e^{(1-\alpha)\log\left[\prod_{i=1}^d(b_i-a_i)\right]}-1}{1-\alpha} = \frac{e^{\log\left[\prod_{i=1}^d(b_i-a_i)\right]^{1-\alpha}}-1}{1-\alpha} = \frac{\left[\prod_{i=1}^d(b_i-a_i)\right]^{1-\alpha}-1}{1-\alpha}. \tag{584}$$

- Eq. (204): Using Eq. (202) and relation (107) we obtain

$$H_{\mathrm{T},\alpha}(\mathbf{y}) = \frac{e^{(1-\alpha)\left(\log\left[(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}\right]-\frac{d\log(\alpha)}{2(1-\alpha)}\right)}-1}{1-\alpha} = \frac{e^{\left(\log\left[(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}\right]^{1-\alpha}-\frac{d\log(\alpha)}{2}\right)}-1}{1-\alpha} \tag{585}$$

$$= \frac{\frac{\left[(2\pi)^{\frac{d}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}\right]^{1-\alpha}}{\alpha^{\frac{d}{2}}}-1}{1-\alpha}. \tag{586}$$

- Eq. (215): Let $f_m(\mathbf{u}) = \frac{\mathbb{I}_{[\mathbf{0},\mathbf{a}_m]}(\mathbf{u})}{\prod_{i=1}^d(\mathbf{a}_m)_i}$ $(m=1,2;\,\mathbf{a}_2 \leq \mathbf{a}_1)$, then

$$[D_L(f_1,f_2)]^2 = \int_{\mathbb{R}^d}[f_1(\mathbf{u})-f_2(\mathbf{u})]^2\,\mathrm{d}\mathbf{u} = \int_{\mathbf{0}}^{\mathbf{a}_1}[f_1(\mathbf{u})-f_2(\mathbf{u})]^2\,\mathrm{d}\mathbf{u} = \int_{\mathbf{0}}^{\mathbf{a}_1} f_1^2(\mathbf{u})-2f_1(\mathbf{u})f_2(\mathbf{u})+f_2^2(\mathbf{u})\mathrm{d}\mathbf{u} \tag{587}$$

$$= \int_{\mathbf{0}}^{\mathbf{a}_1}\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\right]^2\mathrm{d}\mathbf{u} - 2\int_{\mathbf{0}}^{\mathbf{a}_2}\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i}\mathrm{d}\mathbf{u} + \int_{\mathbf{0}}^{\mathbf{a}_2}\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i}\right]^2\mathrm{d}\mathbf{u} \tag{588}$$

$$= \left[\prod_{i=1}^d(\mathbf{a}_1)_i\right]\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\right]^2 - 2\left[\prod_{i=1}^d(\mathbf{a}_2)_2\right]\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i} + \left[\prod_{i=1}^d(\mathbf{a}_2)_i\right]\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i}\right]^2 \tag{589}$$

$$= \frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i} - 2\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i} + \frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i} = \frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i} - \frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}. \tag{590}$$

- Eq. (217): Let $f_m(\mathbf{u}) = \frac{\mathbb{I}_{[\mathbf{0},\mathbf{a}_m]}(\mathbf{u})}{\prod_{i=1}^d(\mathbf{a}_m)_i}$ $(m=1,2;\mathbf{a}_1 \leq \mathbf{a}_2)$, then

$$D_{\mathrm{NB},\alpha}(f_1,f_2) = \int_{\mathbb{R}^d}\left[f_2^\alpha(\mathbf{u}) + \frac{1}{\alpha-1}f_1^\alpha(\mathbf{u}) - \frac{\alpha}{\alpha-1}f_1(\mathbf{u})f_2^{\alpha-1}(\mathbf{u})\right]\mathrm{d}\mathbf{u} \tag{591}$$

$$= \int_{\mathbf{0}}^{\mathbf{a}_2}\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i}\right]^\alpha\mathrm{d}\mathbf{u} + \frac{1}{\alpha-1}\int_{\mathbf{0}}^{\mathbf{a}_1}\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\right]^\alpha\mathrm{d}\mathbf{u} - \frac{\alpha}{\alpha-1}\int_{\mathbf{0}}^{\mathbf{a}_1}\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i}\right]^{\alpha-1}\mathrm{d}\mathbf{u} \tag{592}$$

$$= \left[\prod_{i=1}^d(\mathbf{a}_2)_i\right]\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i}\right]^\alpha + \frac{1}{\alpha-1}\left[\prod_{i=1}^d(\mathbf{a}_1)_i\right]\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\right]^\alpha \tag{593}$$

$$- \frac{\alpha}{\alpha-1}\left[\prod_{i=1}^d(\mathbf{a}_1)_i\right]\frac{1}{\prod_{i=1}^d(\mathbf{a}_1)_i}\left[\frac{1}{\prod_{i=1}^d(\mathbf{a}_2)_i}\right]^{\alpha-1} \tag{594}$$

$$= \left[\prod_{i=1}^d(\mathbf{a}_2)_i\right]^{1-\alpha} + \frac{1}{\alpha-1}\left[\prod_{i=1}^d(\mathbf{a}_1)_i\right]^{1-\alpha} - \frac{\alpha}{\alpha-1}\left[\prod_{i=1}^d(\mathbf{a}_2)_i\right]^{1-\alpha} \tag{595}$$

$$= \left(1-\frac{\alpha}{\alpha-1}\right)\left[\prod_{i=1}^d(\mathbf{a}_2)_i\right]^{1-\alpha} + \frac{1}{\alpha-1}\left[\prod_{i=1}^d(\mathbf{a}_1)_i\right]^{1-\alpha} \tag{596}$$

$$= \frac{1}{\alpha-1}\left(\left[\prod_{i=1}^d(\mathbf{a}_1)_i\right]^{1-\alpha} - \left[\prod_{i=1}^d(\mathbf{a}_2)_i\right]^{1-\alpha}\right). \tag{597}$$

- Eq. (228), (229): The formulas follow from the definitions of $K_{\mathrm{EJR1},u,\alpha}$, $K_{\mathrm{EJR2},u,\alpha}$ [see Eq. (156), (157)] and the analytical formula given in Eq. (232) for $\alpha = 2$.
- Eq. (230), (231): The expressions follow from the definitions of $K_{\mathrm{EJT1},u,\alpha}$, $K_{\mathrm{EJT2},u,\alpha}$ [see Eq. (160), (161)], the analytical formula in (232) for $\alpha = 2$, and the transformation rule of the Rényi and the Tsallis entropy [Eq. (107)].
- Eq. (205): Let $\mathbb{R} \ni y \sim U[a,b]$, $c \geq 1$, then

$$H_{\Phi(u)=u^c,w(u)=\mathbb{I}_{[a,b]}(u)}(y) = \int_{\mathbb{R}}\frac{\mathbb{I}_{[a,b]}(u)}{b-a}\left(\frac{\mathbb{I}_{[a,b]}(u)}{b-a}\right)^c\mathbb{I}_{[a,b]}(u)\mathrm{d}u = \int_a^b\frac{1}{(b-a)^{c+1}}\mathrm{d}u = (b-a)\frac{1}{(b-a)^{c+1}} = \frac{1}{(b-a)^c}. \tag{598}$$

- Eq. (218): Let $f_m(\mathbf{u}) = \frac{\mathbb{I}_{[\mathbf{0},\mathbf{a}_m]}(\mathbf{u})}{\prod_{i=1}^d (\mathbf{a}_m)_i}$ $(m = 1, 2;\ \mathbf{a}_1 \leq \mathbf{a}_2)$, then

$$D_{\chi^2}(f_1, f_2) = \int_{\mathbf{0}}^{\mathbf{a}_2} \frac{\left[\frac{\mathbb{I}_{[\mathbf{0},\mathbf{a}_1]}(\mathbf{u})}{\prod_{i=1}^d (\mathbf{a}_1)_i}\right]^2}{\frac{\mathbb{I}_{[\mathbf{0},\mathbf{a}_2]}(\mathbf{u})}{\prod_{i=1}^d (\mathbf{a}_2)_i}} \mathrm{d}\mathbf{u} - 1 = \int_{\mathbf{0}}^{\mathbf{a}_1} \frac{1}{\prod_{i=1}^d (\mathbf{a}_1)_i^2} \prod_{i=1}^d (\mathbf{a}_2)_i \mathrm{d}\mathbf{u} - 1 = \prod_{i=1}^d (\mathbf{a}_1)_i \frac{1}{\prod_{i=1}^d (\mathbf{a}_1)_i^2} \prod_{i=1}^d (\mathbf{a}_2)_i - 1 \quad (599)$$

$$= \frac{\prod_{i=1}^d (\mathbf{a}_2)_i}{\prod_{i=1}^d (\mathbf{a}_1)_i} - 1. \quad (600)$$

References

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

[2] Ethem Akturk, Baris Bagci, and Ramazan Sever. Is Sharma-Mittal entropy really a step beyond Tsallis and Rényi entropies? Technical report, 2007. http://arxiv.org/abs/cond-mat/0703277.

[3] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, 28:131–142, 1966.

[4] 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.

[5] Shun-Ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2007.

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

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

[8] Neil S. Barnett, Pietro Cerone, Sever Silvestru Dragomir, and A. Sofo. Approximating Csiszár f-divergence by the use of Taylor's formula with integral remainder. *Mathematical Inequalities and Applications*, 5:417–432, 2002.

[9] Michéle Basseville. Divergence measures for statistical data processing - an annotated bibliography. *Signal Processing*, 93:621–633, 2013.

[10] 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.

[11] Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004.

[12] Anil K. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.

[13] 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.

[14] Nils Blomqvist. On a measure of dependence between two random variables. *The Annals of Mathematical Statistics*, 21:593–600, 1950.

[15] Lev M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

[16] Lawrence D. Brown. *Fundamentals of statistical exponential families: with applications in statistical decision theory*. Institute of Mathematical Sciences, Hayworth, CA, USA, 1986.

[17] Jacob Burbea and C. Radhakrishna Rao. On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory*, 28:489–495, 1982.

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

[19] Jean-François Cardoso and Beate Hvam Laheld. Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, 44:3017–3030, 1996.

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

[21] Aiyou Chen. Fast kernel density independent component analysis. In *Independent Component Analysis and Blind Signal Separation (ICA)*, pages 24–31, 2006.

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

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

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

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

[26] Imre Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten. *Publications of the Mathematical Institute of Hungarian Academy of Sciences*, 8:85–108, 1963.

[27] Imre Csiszár. Generalized projections for non-negative functions. *Acta Mathematica Hungarica*, 68:161–185, 1995.

[28] Georges A. Darbellay and Petr Tichavsky. Independent component analysis through direct estimation of the mutual information. In *International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 69–74, 2000.

[29] Georges A. Darbellay and Igor Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 45:1315–1321, 1999.

[30] 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.

[31] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

[32] Ali Dolati and Manuel Úbeda-Flores. On measures of multivariate concordance. *Journal of Probability and Statistical Science*, 4:147–164, 2006.

[33] Daniel R. Dooly, Qi Zhang, Sally A. Goldman, and Robert A. Amar. Multiple-instance learning of real-valued data. *Journal of Machine Learning Research*, 3:651–678, 2002.

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

[35] Dominik M. Endres and Johannes E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49:1858–1860, 2003.

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

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

[38] 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.

[39] Kenji Fukumizu, Francis R. Bach, and Arthur Gretton. Statistical consistency of kernel canonical correlation analysis. *Journal of Machine Learning Research*, 8:361–383, 2007.

[40] Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems (NIPS)*, pages 489–496, 2008.

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

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

[43] Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alexander Smola. Multi-instance kernels. In *International Conference on Machine Learning (ICML)*, pages 179–186, 2002.

[44] Manuel Gil. *On Rényi Divergence Measures for Continuous Alphabet Sources*. PhD thesis, Queen's University, 2011.

[45] 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.

[46] Robert M. Gray. Toeplitz and circulant matrices: A review. *Foundations and Trends in Communications and Information Theory*, 2:155–239, 2006.

[47] Ulf Grenander and Gábor Szegő. *Toeplitz forms and their applications*. University of California Press, 1958.

[48] Arthur Gretton, Karsten M. Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two sample problem. In *Advances in Neural Information Processing Systems (NIPS)*, pages 513–520, 2007.

[49] 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.

[50] 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.

[51] 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.

[52] A.B. Hamza and H. Krim. Jensen-Rényi divergence measure: theoretical and computational perspectives. In *IEEE International Symposium on Information Theory (ISIT)*, page 257, 2003.

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

[54] David Haussler. Convolution kernels on discrete structures. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.

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

[56] Matthias Hein and Olivier Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 136–143, 2005.

[57] 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.

[58] W. Hoeffding. Massstabinvariante korrelationstheorie. *Schriften des Mathematischen Seminars und des Instituts für Angewandte Mathematik der Universität Berlin*, 5:181–233, 1940.

[59] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[60] Marc Van Hulle. Edgeworth approximation of multivariate differential entropy. *Neural Computation*, 17:1903–1910, 2005.

[61] Aapo Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems (NIPS)*, pages 273–279, 1997.

[62] Aapo Hyvärinen. Independent component analysis for time-dependent stochastic processes. In *International Conference on Artificial Neural Networks (ICANN)*, pages 541–546, 1998.

[63] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

[64] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[65] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[66] 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.

[67] Harry Joe. Multivariate concordance. *Journal of Multivariate Analysis*, 35:12–30, 1990.

[68] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz maps into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[69] Christian Jutten and Jeanny Hérault. Blind separation of sources: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[70] 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.

[71] K. Rao Kadiyala and Sune Karlsson. Numerical methods for estimation and inference in bayesian VAR-models. *Journal of Applied Econometrics*, 12:99–132, 1997.

[72] Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabás Póczos, Larry Wasserman, and James Robins. Nonparametric von Mises estimators for entropies, divergences and mutual informations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 397–405, 2015.

[73] Maurice G. Kendall. *Rank correlation methods*. London, Griffin, 1970.

[74] 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.

[75] 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.

[76] Solomon Kullback and Richard Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[77] Jan Kybic. High-dimensional mutual information estimation for image registration. In *International Conference on Image Processing (ICIP)*, pages 1779–1782, 2004.

[78] Russell H. Lambert. *Multichannel Blind Deconvolution: FIR matrix algebra and separation of multipath mixtures*. PhD thesis, University of Southern California, 1996.

[79] Henry Oliver Lancaster. *The Chi-squared Distribution*. John Wiley and Sons Inc, 1969.

[80] Erik Learned-Miller and III. John W. Fisher. ICA using spacings estimates of entropy. *Journal of Machine Learning Research*, 4:1271–1295, 2003.

[81] Nikolai Leonenko, Luc Pronzato, and Vippal Savani. A class of Rényi information estimators for multidimensional densities. *Annals of Statistics*, 36(5):2153–2182, 2008.

[82] 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.

[83] Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37:145–151, 1991.

[84] Weifeng Liu, P.P. Pokharel, and José C. Príncipe. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Transactions on Signal Processing*, 55:5286 – 5298, 2007.

[85] Edward Norton Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20:130–141, 1963.

[86] Russell Lyons. Distance covariance in metric spaces. *Annals of Probability*, 41:3284–3305, 2013.

[87] André F. T. Martins, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Tsallis kernels on measures. In *Information Theory Workshop (ITW)*, pages 298–302, 2008.

[88] André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Nonextensive information theoretical kernels on measures. *Journal of Machine Learning Research*, 10:935–975, 2009.

[89] Marco Massi. A step beyond Tsallis and Rényi entropies. *Physics Letters A*, 338:217–224, 2005.

[90] Jiří Matoušek. On variants of the Johnson-Lindenstrauss lemma. *Random Structures and Algorithms*, 33(2):142–156, 2008.

[91] 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.

[92] Tetsuzo Morimoto. Markov processes and the H-theorem. *Journal of the Physical Society of Japan*, 18:328–331, 1963.

[93] Frederick Mosteller. On some useful "inefficient" statistics. *Annals of Mathematical Statistics*, 17:377–408, 1946.

[94] Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 10–18, 2011.

[95] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29:429–443, 1997.

[96] Roger B. Nelsen. Nonparametric measures of multivariate association. *Lecture Notes-Monograph Series, Distributions with Fixed Marginals and Related Topics*, 28:223–232, 1996.

[97] Roger B. Nelsen. *Distributions with Given Marginals and Statistical Modelling*, chapter Concordance and copulas: A survey, pages 169–178. Kluwer Academic Publishers, Dordrecht, 2002.

[98] Roger B. Nelsen. *An Introduction to Copulas (Springer Series in Statistics)*. Springer, 2006.

[99] Arnold Neumaier and Tapio Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software*, 27(1):27–57, 2001.

[100] 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.

[101] Frank Nielsen and Sylvain Boltz. The Burbea-Rao and Bhattacharyya centroids. *IEEE Transaction on Information Theory*, 57:5455–5466, 2011.

[102] Frank Nielsen and Richard Nock. Entropies and cross-entropies of exponential families. In *IEEE International Conference on Image Processing (ICIP)*, pages 3621–3624, 2010.

[103] Frank Nielsen and Richard Nock. A closed-form expression for the Sharma-Mittal entropy of exponential families. *Journal of Physics A: Mathematical and Theoretical*, 45:032003, 2012.

[104] Frank Nielsen and Richard Nock. On the chi square and higher-order chi distances for approximating f-divergences. *IEEE Signal Processing Letters*, 2:10–13, 2014.

[105] Gang Niu, Wittawat Jitkrittum, Bo Dai, Hirotaka Hachiya, and Masashi Sugiyama. Squared-loss mutual information regularization: A novel information-theoretic approach to semi-supervised learning. In *International Conference on Machine Learning (ICML), JMLR W&CP*, volume 28, pages 10–18, 2013.

[106] Hadi Alizadeh Noughabi and Naser Reza Arghami. A new estimator of entropy. *Journal of Iranian Statistical Society*, 9:53–64, 2010.

[107] Havva Alizadeh Noughabi and Reza Alizadeh Noughabi. On the entropy estimators. *Journal of Statistical Computation and Simulation*, 83:784–792, 2013.

[108] Umut Ozertem, Ismail Uysal, and Deniz Erdogmus. Continuously differentiable sample-spacing entropy estimation. *IEEE Transactions on Neural Networks*, 19:1978–1984, 2008.

[109] Barnabás Páczos, Sergey Krishner, and Csaba Szepesvári. REGO: Rank-based estimation of Rényi information using Euclidean graph optimization. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 605–612, 2010.

[110] Dávid Pál, Barnabás Páczos, 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.

[111] Jason A. Palmer and Scott Makeig. Contrast functions for independent subspace analysis. In *International conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pages 115–122, 2012.

[112] Karl Pearson. On the criterion that a given system of deviations from the probable in the case of correlated system of variables is such that it can be reasonable supposed to have arisen from random sampling. *Philosophical Magazine Series*, 50:157–172, 1900.

[113] 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.

[114] Fernando Pérez-Cruz. Estimation of information theoretic measures for continuous random variables. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1257–1264, 2008.

[115] Barnabás Póczos, Zoubin Ghahramani, and Jeff Schneider. Copula-based kernel dependency measures. In *International Conference on Machine Learning (ICML)*, 2012.

[116] Barnabás Póczos and András Lőrincz. Independent subspace analysis using geodesic spanning trees. In *International Conference on Machine Learning (ICML)*, pages 673–680, 2005.

[117] 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.

[118] Barnabás Póczos and András Lőrincz. Identification of recurrent neural networks by Bayesian interrogation techniques. *Journal of Machine Learning Research*, 10:515–554, 2009.

[119] Barnabás Póczos, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Distribution-free distribution regression. *International Conference on Artificial Intelligence and Statistics (JMLR W&CP)*, 31:507–515, 2013.

[120] Barnabás Póczos and Jeff Schneider. On the estimation of α-divergences. In *International conference on Artificial Intelligence and Statistics (AISTATS)*, pages 609–617, 2011.

[121] 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.

[122] Barnabás Póczos, Zoltán Szabó, and Jeff Schneider. Nonparametric divergence estimators for independent subspace analysis. In *European Signal Processing Conference (EUSIPCO)*, pages 1849–1853, 2011.

[123] Barnabás Póczos, Bálint Takács, and András Lőrincz. Independent subspace analysis on innovations. In *European Conference on Machine Learning (ECML)*, pages 698–706, 2005.

[124] 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.

[125] Barnabás Póczos, Liang Xiong, Dougal Sutherland, and Jeff Schneider. Support distribution machines. Technical report, Carnegie Mellon University, 2012. http://arxiv.org/abs/1202.0302.

[126] Ravikiran Rajagopal and Lee C. Potter. Multivariate MIMO FIR inverses. *IEEE Transactions on Image Processing*, 12:458–465, 2003.

[127] David Ramírez, Javier Vía, Ignacio Santamaría, and Pedro Crespo. Entropy and Kullback-Leibler divergence estimation based on Szegő's theorem. In *European Signal Processing Conference (EUSIPCO)*, pages 2470–2474, 2009.

[128] Murali Rao, Sohan Seth, Jianwu Xu, Yunmei Chen, Hemant Tagare, and José C. Príncipe. A test of independence based on a generalized correlation function. *Signal Processing*, 91:15–27, 2011.

[129] Soumya Ray and David Page. Multiple instance regression. In *International Conference on Machine Learning*, pages 425–432, 2001.

[130] Alfréd Rényi. On measures of information and entropy. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1961.

[131] Alfréd Rényi. *Probability theory*. Elsevier, 1970.

[132] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method*. Springer, 2004.

[133] Marco Scarsini. On measures of concordance. *Stochastica*, 8:201–218, 1984.

[134] Friedrich Schmid and Rafael Schmidt. Multivariate conditional versions of Spearman's rho and related measures of tail dependence. *Journal of Multivariate Analysis*, 98:1123–1140, 2007.

[135] Friedrich Schmid and Rafael Schmidt. Multivariate extensions of Spearman's rho and related statistics. *Statistics & Probability Letters*, 77:407–416, 2007.

[136] 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.

[137] 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.

[138] B. Schweizer and Edward F. Wolff. On nonparametric measures of dependence for random variables. *The Annals of Statistics*, 9:879–885, 1981.

[139] Dino Sejdinovic, Arthur Gretton, and Wicher Bergsma. A kernel test for three-variable interactions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1124–1132, 2013.

[140] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *Annals of Statistics*, 41:2263–2291, 2013.

[141] Sohan Seth and José C. Príncipe. Compressed signal reconstruction using the correntropy induced metric. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3845 – 3848, 2008.

[142] 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.

[143] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[144] Bhudev D. Sharma and Dharam P. Mittal. New nonadditive measures of inaccuracy. *Journal of Mathematical Sciences*, 10:122–133, 1975.

[145] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[146] Masaaki Sibuya. Bivariate extreme statistics. *Annals of the Institute of Statistical Mathematics*, 11:195–210, 1959.

[147] 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.

[148] A. Sklar. Fonctions de répartition á n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:229–231, 1959.

[149] Kai-Sheng Song. Rényi information, loglikelihood and an intrinsic distribution measure. *Journal of Statistical Planning and Inference*, 93:51–69, 2001.

[150] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15:72–101, 1904.

[151] 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.

[152] 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.

[153] Dan Stowell and Mark D. Plumbley. Fast multidimensional entropy estimation by k-d partitioning. *IEEE Signal Processing Letters*, 16:537–540, 2009.

[154] Milan Studený and Jirina Vejnarová. The multiinformation function as a tool for measuring stochastic dependence. In *Learning in Graphical Models*, pages 261–296, 1998.

[155] Taiji Suzuki, Masashi Sugiyama, Takafumi Kanamori, and Jun Sese. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10:S52, 2009.

[156] Zoltán Szabó. Complete blind subspace deconvolution. In *International Conference on Independent Component Analysis and Signal Separation (ICA)*, pages 138–145, 2009.

[157] 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.

[158] Zoltán Szabó. Information theoretical estimators toolbox. *Journal of Machine Learning Research*, 15:283–287, 2014.

[159] Zoltán Szabó, Arthur Gretton, Barnabás Póczos, and Bharath Sriperumbudur. Two-stage sampled learning theory on distributions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, San Diego, California, USA, 9-12 May 2015. http://arxiv.org/abs/1402.1754.

[160] 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.

[161] Zoltán Szabó and András Lőrincz. Towards independent subspace analysis in controlled dynamical systems. In *ICA Research Network International Workshop (ICARN)*, pages 9–12, 2008.

[162] Zoltán Szabó and András Lőrincz. Complex independent process analysis. *Acta Cybernetica*, 19:177–190, 2009.

[163] 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.

[164] Zoltán Szabó and András Lőrincz. Distributed high dimensional information theoretical image registration via random projections. *Digital Signal Processing*, 22(6):894–902, 2012.

[165] Zoltán Szabó and Barnabás Póczos. Nonparametric independent process analysis. In *European Signal Processing Conference (EUSIPCO)*, pages 1718–1722, 2011.

[166] 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.

[167] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Undercomplete blind subspace deconvolution. *Journal of Machine Learning Research*, 8:1063–1095, 2007.

[168] 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.

[169] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Auto-regressive independent process analysis without combinatorial efforts. *Pattern Analysis and Applications*, 13:1–13, 2010.

[170] Zoltán Szabó, Barnabás Póczos, and András Lőrincz. Separation theorem for independent subspace analysis and its consequences. *Pattern Recognition*, 45:1782–1791, 2012.

[171] 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.

[172] Gábor J. Székely and Maria L. Rizzo. Testing for equal distributions in high dimension. *InterStat*, 5, 2004.

[173] Gábor J. Székely and Maria L. Rizzo. A new test for multivariate normality. *Journal of Multivariate Analysis*, 93:58–80, 2005.

[174] Gábor J. Székely and Maria L. Rizzo. Brownian distance covariance. *The Annals of Applied Statistics*, 3:1236–1265, 2009.

[175] 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.

[176] Anisse Taleb and Christian Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on Signal Processing*, 10(47):2807–2820, 1999.

[177] M. D. Taylor. Multivariate measures of concordance. *Annals of the Institute of Statistical Mathematics*, 59:789–806, 2007.

[178] 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.

[179] Fabian J. Theis. Towards a general independent subspace analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1361–1368, 2007.

[180] Flemming Topsøe. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, 46:1602–1609, 2000.

[181] Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52:479–487, 1988.

[182] Manuel Úbeda-Flores. Multivariate versions of Blomqvist's beta and Spearman's footrule. *Annals of the Institute of Statistical Mathematics*, 57:781–788, 2005.

[183] 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.

[184] 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.

[185] Oldrich Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society, Series B*, 38:54–59, 1976.

[186] T. Villmann and S. Haase. Mathematical aspects of divergence based vector quantization using Fréchet-derivatives. Technical report, University of Applied Sciences Mittweida, 2010.

[187] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.

[188] 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.

[189] Fei Wang, Tanveer Syeda-Mahmood, Baba C. Vemuri, David Beymer, and Anand Rangarajan. Closed-form Jensen-Rényi divergence for mixture of Gaussians and applications to group-wise shape registration. *Medical Image Computing and Computer-Assisted Intervention*, 12:648–655, 2009.

[190] 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.

[191] 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.

[192] Zhuang Wang, Liang Lan, and Slobodan Vucetic. Mixture model for multiple instance regression and applications in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 50:2226–2237, 2012.

[193] Satori Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4:66–82, 1960.

[194] Edward F. Wolff. N-dimensional measures of dependence. *Stochastica*, 4:175–188, 1980.

[195] 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.

[196] Joseph E. Yukich. Probability theory of classical Euclidean optimization problems. *Lecture Notes in Mathematics*, 1675, 1998.

[197] Wojciech Zaremba, Arthur Gretton, and Matthew Blaschko. B-tests: Low variance kernel two-sample tests. In *Advances in Neural Information Processing Systems (NIPS)*, pages 755–763, 2013.

[198] 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.

[199] V. M. Zolotarev. Probability metrics. *Theory of Probability and its Applications*, 28:278–302, 1983.