Information Theoretical Estimators (ITE) Toolbox in Python, v1.1

Zoltán Szabó

August 12, 2020

Contents

List of Examples

1 Introduction

Measuring the uncertainty, independence, association, inner product or distance of random variables is a central problem in machine learning and statistics with numerous applications. Despite the large number of successful applications and new opportunities, there are quite few packages which allow the systematic exploration and comparison of information theoretical measures.[1] To alleviate this serious bottleneck we created this *Information Theoretical Estimators (ITE)* toolbox. It

1. is the redesigned, Python implementation of the Matlab/Octave ITE toolbox[2].
2. can estimate numerous entropy, mutual information, divergence, association measures, cross quantities, and kernels on distributions (see the list below).
3. can be used to solve information theoretical optimization problems in a high-level way.
4. comes with several demos.

Some details:

- **Estimated quantities**:
  - **entropy**: Shannon entropy, Rényi entropy, Tsallis entropy (Havrda and Charvát entropy), Sharma-Mittal entropy, Φ-entropy (f-entropy).
  - **mutual information**: Shannon mutual information (total correlation, multi-information), Rényi mutual information, Tsallis mutual information, χ^2 mutual information (squared-loss mutual information, mean square contingency), L_2 mutual information, copula-based kernel dependency, kernel canonical correlation analysis (KCCA), kernel generalized variance (KGV), multivariate version of Hoeffding's Φ, Hilbert-Schmidt independence criterion (HSIC), distance covariance, distance correlation, Lancaster three-variable interaction.
  - **divergence**: Kullback-Leibler divergence (relative entropy, I directed divergence), Rényi divergence, Tsallis divergence, Sharma-Mittal divergence, Pearson χ^2 divergence (χ^2 distance), Hellinger distance, L_2 divergence, f-divergence (Csiszár-Morimoto divergence, Ali-Silvey distance), maximum mean discrepancy (MMD; kernel distance, current distance), energy distance (N-distance; specifically the Cramer-Von Mises distance), Bhattacharyya distance, non-symmetric Bregman distance (Bregman divergence), symmetric Bregman distance, J-distance (symmetrised Kullback-Leibler divergence, J divergence), K divergence, L divergence, Jensen-Shannon divergence, Jensen-Rényi divergence, Jensen-Tsallis divergence.
  - **association measures**: multivariate extensions of Spearman's ρ (Spearman's rank correlation coefficient, grade correlation coefficient), multivariate conditional version of Spearman's ρ, lower and 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; specific convolution kernel), probability product kernel, Bhattacharyya kernel (Bhattacharyya coefficient, Hellinger affinity), Jensen-Shannon kernel, Jensen-Tsallis kernel, exponentiated Jensen-Shannon kernel, exponentiated Jensen-Rényi kernels, exponentiated Jensen-Tsallis kernels.
  - **conditional entropy**: conditional Shannon entropy.
  - **conditional mutual information**: conditional Shannon mutual information.
- **Web**: `https://bitbucket.org/szzoli/ite-in-python/`. Comments are welcome.
- **Follow ITE**: on
  - Bitbucket (`https://bitbucket.org/szzoli/ite-in-python/follow`),
  - Twitter (`https://twitter.com/ITEtoolbox`).
- **Publications/applications**: Papers using ITE are collected at `https://bitbucket.org/szzoli/ite-in-python/wiki`.
- **Author**: Zoltán Szabó (`http://www.cmap.polytechnique.fr/~zoltan.szabo/`).

[1] A few nice examples focusing on discrete variables or specialized applications and methods are `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`, `http://cran.r-project.org/web/packages/entropy/`, `https://github.com/dit/dit`, `https://pypi.python.org/pypi/universal-divergence/0.2.0`, `https://github.com/baccuslab/shannon`, or `http://fr.mathworks.com/matlabcentral/fileexchange/35625-information-theory-toolbox`.

[2] See `https://bitbucket.org/szzoli/ite/`. In the sequel we will use 'Matlab ITE' instead of 'Matlab/Octave ITE'.

- **Citing**: If you use the ITE toolbox in your work, please cite it [6].[3] The source code also contains references for the individual methods and the quantities estimated.

- **License**: GPLv3(>=).

- **Requirements**:

  1. Python 3, SciPy [∋(typically) NumPy, Matplotlib].[4] You can get these tools by pip[5] (∈ Python 3 ≥ 3.4).
     - The system-wide installation is as follows:

       ```
       # python3 -m pip install scipy      # '#' denotes bash prompt (with root rights)
       # python3 -m pip install numpy      # if you do not get it by SciPy
       # python3 -m pip install matplotlib # -||- (:=same comment)
       ```

     - The user-specific installation is

       ```
       > python3 -m pip install --user scipy  # '>' stands for the bash prompt (with normal user
       > python3 -m pip install --user numpy  # rights)
       > python3 -m pip install --user matplotlib
       ```

  2. Nose, IPython: optional.[6]

  Note: Installing Anaconda gives all these tools, with Intel MKL (Math Kernel Library).[7]

The rest of the documentation is structured as follows:

- Section 2 is about the installation of ITE, how to import it and run its built-in demos, and examples of usage. Section 3 enlists the definitions of the estimated quantities.

- Section A is for developers with details on (i) the directory structure of the toolbox, (ii) how to add new estimators and run doctests, (iii) parameter passing in (certain) meta estimators. The correspondence between Python and Matlab ITE is detailed in Section B. Section C contains the axiomatic formulation of concordance and dependence.

2 Getting Started: Installation, Built-in Demos, Examples

This section is about the installation and importing of the ITE toolbox, running its built-in demos, followed examples.

- Installation: download the ITE archive (https://bitbucket.org/szzoli/ite_in_python/downloads), extract its contents. We will denote the resulting main folder (containing demos, doc, ite, LICENSE.txt, ...) as ite.

- Start a working session:

  ```
  > ipython3          # see the first bullet point of the note below
  >>> import ite      # change first to the ite directory, if it is not on your Python path;
                      # '>>>' denotes the prompt in the (I)Python console
  ```

  Note:

  - Throughout this documentation for simplicity/efficiency I assume that you use IPython; you might want to do this implicitly via an IDE such as PyCharm[8].
  - You can add the ITE package to the Python path by

    ```
    >>> import sys
    >>> sys.path.insert(1,'/path/to/directory/containing/ite')
    ```

[3] .bib: http://www.cmap.polytechnique.fr/~zoltan.szabo/ITE.bib.
[4] See https://www.python.org/ and http://www.scipy.org/.
[5] See https://pypi.python.org/pypi/pip.
[6] See http://nose.readthedocs.io/en/latest/ and https://ipython.org/.
[7] See https://www.continuum.io/downloads.
[8] See https://www.jetbrains.com/pycharm/.

- Running the built-in demos (see Table 9): Change to the ite/demos/analytical_values directory and run the demos. Example:

```
>>> run demo_h_shannon  # run ∈ IPython; notice that the '.py' extension could be discarded
```

- Examples: In the first example we estimate the Shannon entropy of a random variable $\mathbf{y}$ [$H(\mathbf{y})$, see (1)] using the k-nearest neighbor method; the estimator is called BHShannonKnnK in ITE. $\mathbf{y}$ will be uniformly distributed on the 3-dimensional unit cube ($\in [0,1]^3$; $d = 3$) from which we have $T = 1000$ samples. The first estimator ($co1$ below) relies on the default parameter setting, the second one ($co2$) is based on user-specified parameters. Particularly, in the second case we specify the kNN computation method, the number of neighbors (k) and allow approximation in the kNN phase (*eps*; to speed up computation). For alternative entropy estimators, see Table 1.

  Example 1 (Entropy estimation)

```
>>> import ite                          # import the ITE toolbox (1x)
>>> from numpy.random import rand       # we will use 'rand' to create the observations
>>> co1 = ite.cost.BHShannon_KnnK()     # initialize the entropy (2nd character = 'H') estimator
>>> print(co1)                          # print estimator-1
>>> y = rand(1000, 3)                   # size: number of samples × dimension, $\{\mathbf{y}_t\}_{t=1}^{1000}$, $\mathbf{y}_t \in \mathbb{R}^3$
>>> h = co1.estimation(y)               # entropy estimation
>>>
>>> co2 = ite.cost.BHShannon_KnnK(knn_method='cKDTree', k=2, eps=0.1) # with other estimator
                                                                      # parameters
>>> print(co2)                        # print estimator-2
>>> h2 = co2.estimation(y)            # entropy estimation
```

  In our second example we consider the estimation of the classical Shannon mutual information [see (6)]. The random variable ($\mathbf{y}$) of interest is partitioned into 3 blocks: $\mathbf{y} = \left[\mathbf{y}^1; \mathbf{y}^2; \mathbf{y}^3\right]$ ($\mathbf{y}^1 \in \mathbb{R}^2$, $\mathbf{y}^2 \in \mathbb{R}^3$, $\mathbf{y}^3 \in \mathbb{R}^4$) and we want to estimate $I(\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3)$, the mutual information of $\mathbf{y}^m$-s. $T = 2000$ samples are used for estimation. We estimate the mutual information of $\mathbf{y}^m$-s from Kullback-Leibler divergence [see (6) and (24)]. These type of *derived* estimators are called meta estimators in ITE (MIShannon_DKL; 1st character = 'M'). 'Base' estimators refer to non-derived ones, as it was the case in Example 1 (BHShannon_KnnK: 1st character = 'B').

  Example 2 (Mutual information estimation (association measure: similarly))

```
>>> from numpy.random import randn # we will use 'randn' to create the observations
>>> from numpy import array        # an 'array' will contain the subspace dimensions: $[d_1; d_2; d_3]$
>>> co = ite.cost.MIShannon_DKL()  # initialize the mutual information estimator
                                   # (MIShannon_DKL: 2nd character = 'I')
>>> ds = array([2, 3, 4])          # $\mathbf{y}^1 \in \mathbb{R}^2$, $\mathbf{y}^2 \in \mathbb{R}^3$, $\mathbf{y}^3 \in \mathbb{R}^4$, $d = d_1 + d_2 + d_3 = 2 + 3 + 4 = 9$
>>> t = 2000                       # number of samples
>>> y = randn(t, sum(ds))          # size: number of samples × dimension
>>> i = co.estimation(y, ds)       # estimate mutual information
```

  Alternative mutual information measures/techniques are listed in Table 2. Estimation of association measures is analogous, the available methods are covered in Table 4.

  In our third example we estimate $D(\mathbf{y}^1, \mathbf{y}^2)$, the Kullback-Leibler divergence between two random quantities $\mathbf{y}^1$ and $\mathbf{y}^2$ [see (24)[9]], via k-nearest neighbors (BDKL_KnnK). We have $T_1 = 2000$ samples from $\mathbf{y}^1$ and $T_2 = 3000$ samples from $\mathbf{y}^2$.

  Example 3 (Divergence estimation (cross quantity, kernel on distributions: analogously))

[9] We identify random variables with their distributions or their pdf-s (probability density functions; meant w.r.t. the Lebesgue measure). Many of the information theoretical quantities can be formulated more generally, but to keep the presentation simple we will avoid going into measure theoretical details.

```
>>> from numpy.random import randn  # 'randn' is used to generate our observations
>>> co = ite.cost.BDKL_KnnK()       # initialize the divergence (2nd character = 'D') estimator
>>> dim = 3                         # $\mathbf{y}^1 \in \mathbb{R}^3$, $\mathbf{y}^2 \in \mathbb{R}^3$
>>> t1, t2 = 2000, 3000             # number of samples from $\mathbf{y}^1$ and $\mathbf{y}^2$
>>> y1 = randn(t1, dim)             # size: number of samples1 × dimension, $\{\mathbf{y}_t^1\}$
>>> y2 = randn(t2, dim)             # size: number of samples2 × dimension, $\{\mathbf{y}_t^2\}$
>>> d = co.estimation(y1, y2)       # estimate KL divergence
```

For other divergence measures or estimators see Table 3. The estimation of cross quantities and kernels on distributions (Table 5, Table 6) can be carried out in the same way.

In our fourth example, we focus on the estimation of the conditional Shannon entropy of $\mathbf{y}^1$ given $\mathbf{y}^2$ [see (68)]. We have $T = 5000$ samples from the joint distribution of $\mathbf{y} = \left[\mathbf{y}^1, \mathbf{y}^2\right]$; it is assumed to be Gaussian in this example.

Example 4 (Conditional entropy estimation)

```
>>> from numpy import dot                                 # create observations
>>> from numpy.random import rand, multivariate_normal   # -||-
>>> dim1, dim2 = 1, 2                                     # $\mathbf{y}^1 \in \mathbb{R}^1$, $\mathbf{y}^2 \in \mathbb{R}^2$
>>> dim = dim1 + dim2                                     # $\mathbf{y} = [\mathbf{y}^1, \mathbf{y}^2] \in \mathbb{R}^{1+2=3}$
>>> t = 5000                                              # number of samples
>>> co = ite.cost.BcondHShannon_HShannon() # initialize the conditional entropy ('condH')
                                           # estimator
>>> m, l = rand(dim), rand(dim, dim)       # mean ($\mathbf{m}$)
>>> c = dot(l, l.T)                        # covariance ($\mathbf{\Sigma}$), $\mathbf{y} = N(\mathbf{m}, \mathbf{\Sigma})$
>>> y = multivariate_normal(m, c, t)       # $\{\mathbf{y}_t\}_{t=1}^{5000}$, $\mathbf{y}_t = \left[\mathbf{y}_t^1, \mathbf{y}_t^2\right] \in \mathbb{R}^3$
>>> cond_h = co.estimation(y, dim1)        # estimate conditional entropy
```

In our fifth example, the task is to estimate the conditional Shannon mutual information of $\mathbf{y}^1$ and $\mathbf{y}^2$ given $\mathbf{y}^3$ [see (69)]. We are given $T = 3000$ samples from the joint distribution of $\mathbf{y} = \left[\mathbf{y}^1; \mathbf{y}^2; \mathbf{y}^3\right]$; in the example below it is Gaussian.

Example 5 (Conditional mutual information estimation)

```
>>> from numpy import dot, array                          # create observations
>>> from numpy.random import rand, multivariate_normal   # -||-
>>> ds = array([1, 2, 3])                                 # $\mathbf{y}^1 \in \mathbb{R}$, $\mathbf{y}^2 \in \mathbb{R}^2$, $\mathbf{y}^3 \in \mathbb{R}^3$
>>> dim = sum(ds)                                         # $d = d_1 + d_2 + d_3 = 1 + 2 + 3 = 6$,
                                                          # $\mathbf{y} = \left[\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\right] \in \mathbb{R}^6$
>>> t = 3000                                              # number of samples
>>> co = ite.cost.BcondIShannon_HShannon() # initialize the conditional mutual information
                                           # ('condI') estimator
>>> m, l = rand(dim), rand(dim, dim)       # mean ($\mathbf{m}$)
>>> c = dot(l, l.T)                        # covariance ($\mathbf{\Sigma}$), $\mathbf{y} = N(\mathbf{m}, \mathbf{\Sigma})$
>>> y = multivariate_normal(m, c, t)       # $\{\mathbf{y}_t\}_{t=1}^{3000}$, $\mathbf{y}_t = \left[\mathbf{y}_t^1, \mathbf{y}_t^2, \mathbf{y}_t^3\right]$
>>> cond_i = co.estimation(y, ds)          # estimate conditional mutual information
```

3 Estimated Quantities and Estimators

In this section we give the definitions of the information theoretical quantities implemented in ITE: Section 3.1 focuses on unconditional quantities, Section 3.2 contains the conditional ones. Section 3.3 is about their estimators.

3.1 Unconditional Quantities

This part is structured as follows: entropy (Section 3.1.1), mutual information (Section 3.1.2), divergence (Section 3.1.3), association measure (Section 3.1.4) cross quantity (Section 3.1.5), kernel on distributions (Section 3.1.6).

3.1.1 Entropy

- Notation: $\mathbb{R}^d \ni \mathbf{y} \sim f$, in other words the d-dimensional random variable $\mathbf{y}$ has density f.
- Goal: We want to estimate the entropy of $\mathbf{y} \in \mathbb{R}^d$ from which we have i.i.d. (independent identically distributed) samples, $\{\mathbf{y}_t\}_{t=1}^T$ ($\mathbf{y}_t \in \mathbb{R}^d$, $t = 1, \dots T$).
- Definitions: The Shannon entropy (H), Rényi entropy ($H_{\mathrm{R},\alpha}$), Tsallis entropy ($H_{\mathrm{T},\alpha}$; also called Havrda and Charvát entropy), Sharma-Mittal entropy ($H_{\mathrm{SM},\alpha,\beta}$), Φ-entropy ($H_{\Phi,w}$; f-entropy[10]) are defined as[11]

$$H(\mathbf{y}) = -\int_{\mathbb{R}^d} f(\mathbf{u}) \log f(\mathbf{u}) \mathrm{d}\mathbf{u}, \tag{1}$$

$$H_{\mathrm{R},\alpha}(\mathbf{y}) = \frac{1}{1-\alpha} \log \int_{\mathbb{R}^d} f^\alpha(\mathbf{u}) \mathrm{d}\mathbf{u}, \quad (\alpha \neq 1) \qquad \lim_{\alpha \to 1} H_{\mathrm{R},\alpha} = H, \tag{2}$$

$$H_{\mathrm{T},\alpha}(\mathbf{y}) = \frac{1}{\alpha-1}\left[1 - \int_{\mathbb{R}^d} f^\alpha(\mathbf{u}) \mathrm{d}\mathbf{u}\right] = \frac{e^{(1-\alpha)H_{\mathrm{R},\alpha}(\mathbf{y})} - 1}{1-\alpha}, \quad (\alpha \neq 1) \qquad \lim_{\alpha \to 1} H_{\mathrm{T},\alpha} = H, \tag{3}$$

$$H_{\mathrm{SM},\alpha,\beta}(\mathbf{y}) = \frac{1}{1-\beta}\left[\left(\int_{\mathbb{R}^d} f^\alpha(\mathbf{u}) \mathrm{d}\mathbf{u}\right)^{\frac{1-\beta}{1-\alpha}} - 1\right], \quad (\alpha > 0, \alpha \neq 1, \beta \neq 1), \tag{4}$$

$$H_{\Phi,w}(y) = \int_{\mathbb{R}} f(u)\Phi(f(u))w(u) \mathrm{d}u, \ (\mathbb{R} \ni y \sim f). \tag{5}$$

- Note:
  - $H_{\mathrm{SM},\alpha,\beta}$: $\lim_{\beta \to 1} H_{\mathrm{SM},\alpha,\beta}(\mathbf{y}) = H_{\mathrm{R},\alpha}(\mathbf{y})$, $H_{\mathrm{SM},\alpha,\alpha}(\mathbf{y}) = H_{\mathrm{T},\alpha}(\mathbf{y})$, $\lim_{(\alpha,\beta)\to(1,1)} H_{\mathrm{SM},\alpha,\beta}(\mathbf{y}) = H(\mathbf{y})$.

3.1.2 Mutual Information

- Notations: $\mathbb{R}^d \ni \mathbf{y} = \left[\mathbf{y}^1; \dots; \mathbf{y}^M\right] \sim f$, $\mathbb{R}^{d_m} \ni \mathbf{y}^m \sim f_m$ ($d = \sum_{m=1}^M d_m$). f_S ($S \subseteq \{1, \dots, M\}$) stands for the associated marginals; for example $f_{\{1,2\}}$ is the density function of $\left[\mathbf{y}^1; \mathbf{y}^2\right]$. $\widetilde{\mathbf{y}^m}$ denotes an identically distributed copy of $\mathbf{y}^m$. '$\perp\!\!\!\perp$' means independence, '$\vee$' denotes the logical 'or'. $\mathbb{E}$ is for expectation, cov is covariance, var denotes variance, $i = \sqrt{-1}$, $\langle \cdot, \cdot \rangle_2$ is the Euclidean inner product, $\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} \mathrm{d}x$ is the Gamma function. Let us define the weight function $w\left(\mathbf{u}^1, \mathbf{u}^2\right) = \frac{1}{c(d_1,\alpha)c(d_2,\alpha)\left[\|\mathbf{u}^1\|_2\right]^{d_1+\alpha}\left[\|\mathbf{u}^2\|_2\right]^{d_2+\alpha}}$ with $\alpha \in (0,2)$, where $c(d,\alpha) = \frac{2\pi^{\frac{d}{2}}\Gamma\left(1-\frac{\alpha}{2}\right)}{\alpha 2^\alpha \Gamma\left(\frac{d+\alpha}{2}\right)}$. $\varphi_{12}\left(\mathbf{u}^1, \mathbf{u}^2\right) = \mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\left[e^{i\left\langle \mathbf{u}^1, \mathbf{y}^1 \right\rangle + i\left\langle \mathbf{u}^2, \mathbf{y}^2 \right\rangle}\right]$, $\varphi_j\left(\mathbf{u}^j\right) = \mathbb{E}_{\mathbf{y}^j}\left[e^{i\left\langle \mathbf{u}^j, \mathbf{y}^j \right\rangle}\right]$, $(j = 1, 2)$ are the characteristic functions of $\left[\mathbf{y}^1; \mathbf{y}^2\right]$, $\mathbf{y}^1$ and $\mathbf{y}^2$. Given a reproducing kernel k, $\mathcal{H}(k)$ is the associated RKHS (reproducing kernel Hilbert space). $\otimes_{i=1}^3 \mathcal{H}(k_i)$ denotes the tensor product of $\mathcal{H}(k_i)$-s. $\mu_q = \int k(\cdot, u) q(u) \mathrm{d}u = \mathbb{E}_{u \sim q}\left[k(\cdot, u)\right]$ is the mean embedding of q to $\mathcal{H}(k)$; q is often a pdf [see (20) and (21) for an example where q is not a pdf; it can be negative]. F: cdf (cumulative density function) of $\mathbf{y} = \left[y^1; \dots; y^d\right]$, F_i: cdf of y^i. C: copula of $\mathbf{y}$, i.e. $F(\mathbf{y}) = C\left(F_1\left(y^1\right), \dots, F_d\left(y^d\right)\right)$, in other words $C(\mathbf{u}) = \mathbb{P}(\mathbf{U} \leq \mathbf{u})$ where $\mathbf{U} = \left[F_1\left(y^1\right); \dots; F_d\left(y^d\right)\right] \in [0,1]^d$. $\Pi(u_1, \dots, u_d) = \prod_{i=1}^d u_i$ is the product copula. f_U: uniform density on $[0,1]^M$.
- Goal: We consider the estimation of the mutual information of the d_m-dimensional components ($\mathbf{y}^m$) of the random variable $\mathbf{y}$ using an i.i.d. sample set $\{\mathbf{y}_t\}_{t=1}^T$ from $\mathbf{y}$.
- Definitions: The Shannon mutual information (I; also known as total correlation or multi-information), Rényi mutual information ($I_{\mathrm{R},\alpha}$), Tsallis mutual information ($I_{\mathrm{T},\alpha}$), χ^2 mutual information (I_{χ^2}; for $M = 2$ also called squared-loss mutual information; mean square contingency $= \sqrt{I_{\chi^2}}$), L_2 mutual information (I_{L_2}), copula-based kernel dependency (I_{c}), kernel canonical correlation analysis (I_{KCCA}; KCCA), kernel generalized variance (I_{KGV}; KGV), multivariate version of Hoeffding's Φ (I_Φ), Hilbert-Schmidt independence criterion (I_{HSIC}; HSIC), distance covariance (I_{dCov}), distance correlation (I_{dCor}), Lancaster 3-variable interaction ($I_{\text{3-Lanc}}$), three-variable joint independence measure ($I_{\text{3-joint}}$) are defined as

$$I\left(\mathbf{y}^1, \dots, \mathbf{y}^M\right) = \int_{\mathbb{R}^d} f\left(\mathbf{u}^1, \dots, \mathbf{u}^M\right) \log\left[\frac{f\left(\mathbf{u}^1, \dots, \mathbf{u}^M\right)}{\prod_{m=1}^M f_m(\mathbf{u}^m)}\right] \mathrm{d}\mathbf{u}^1 \cdots \mathrm{d}\mathbf{u}^M, \tag{6}$$

[10] Since f also denotes the density in (5), we refer to the quantity as the Φ-entropy.

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

$$= D\left(f, \prod_{m=1}^{M} f_m\right) \qquad \text{[see (24) for the definition of } D\text{]}, \tag{7}$$

$$I_{\mathrm{R},\alpha}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{R},\alpha}\left(f, \prod_{m=1}^{M} f_m\right), \text{ for } D_{\mathrm{R},\alpha}\text{, see (25)}, \tag{8}$$

$$I_{\mathrm{T},\alpha}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{T},\alpha}\left(f, \prod_{m=1}^{M} f_m\right), \text{ for } D_{\mathrm{T},\alpha}\text{, see (26)}, \tag{9}$$

$$I_{\chi^2}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\chi^2}\left(f, \prod_{m=1}^{M} f_m\right), \text{ for } D_{\chi^2}\text{, see (28)}, \tag{10}$$

$$I_{\mathrm{L}_2}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = D_{\mathrm{L}_2}\left(f, \prod_{m=1}^{M} f_m\right), \text{ for } D_{\mathrm{L}_2}\text{, see (30)}, \tag{11}$$

$$I_{\mathrm{c}}\left(y^1,\ldots,y^M\right) = D_{\mathrm{MMD}}(f_{\mathbf{z}}, f_U), \text{ for } D_{\mathrm{MMD}} \text{ see (32)}, \qquad \mathbf{z} = \left[F_1\left(y^1\right);\ldots;F_M\left(y^M\right)\right] \in \mathbb{R}^M, \tag{12}$$

$$I_{\mathrm{KCCA}}(\mathbf{y}^1,\mathbf{y}^2) = \sup_{g_1\in\mathcal{H}(k_1),g_2\in\mathcal{H}(k_2)} \frac{\mathrm{cov}[g_1(\mathbf{y}^1),g_2(\mathbf{y}^2)]}{\sqrt{\mathrm{var}\left[g_1(\mathbf{y}^1)\right]+\kappa\left\|g_1\right\|^2_{\mathcal{H}(k_1)}}\sqrt{\mathrm{var}\left[g_2(\mathbf{y}^2)\right]+\kappa\left\|g_2\right\|^2_{\mathcal{H}(k_2)}}}, \quad (\kappa>0), \tag{13}$$

$$I_{\mathrm{KGV}}\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right) = -\frac{1}{2}\log\left[\frac{\det(\mathbf{C})}{\prod_{m=1}^{M}\det(\mathbf{C}^{m,m})}\right], \ \mathbf{C} = \left[\mathbf{C}^{i,j}\right], \ \varphi(\mathbf{y}) := \left[\varphi_m\left(\mathbf{y}^m\right)\right]_{m=1}^{M}, \tag{14}$$

$$\mathbf{C}^{i,j} = \mathrm{cov}\left[\varphi_i\left(\mathbf{y}^i\right),\varphi_j\left(\mathbf{y}^j\right)\right], \tag{15}$$

$$I_{\Phi}\left(y^1,\ldots,y^d\right) = I_{\Phi}(C) = \left(h_2(d)\int_{[0,1]^d}[C(\mathbf{u})-\Pi(\mathbf{u})]^2\mathrm{d}\mathbf{u}\right)^{\frac{1}{2}}, \tag{16}$$

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

$$I_{\mathrm{HSIC}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \left\|C_{\mathbf{y}^1\mathbf{y}^2}\right\|^2_{\mathrm{HS}}, \qquad C_{\mathbf{y}^1\mathbf{y}^2} = \mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\left(\left[k_1\left(\cdot,\mathbf{y}^1\right)-\boldsymbol{\mu}_1\right]\otimes\left[k_2\left(\cdot,\mathbf{y}^2\right)-\boldsymbol{\mu}_2\right]\right), \tag{17}$$

$$I_{\mathrm{dCov}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \left\|\varphi_{12}-\varphi_1\varphi_2\right\|_{L^2_w} = \sqrt{\int_{\mathbb{R}^{d_1+d_2}}\left|\varphi_{12}\left(\mathbf{u}^1,\mathbf{u}^2\right)-\varphi_1\left(\mathbf{u}^1\right)\varphi_2\left(\mathbf{u}^2\right)\right|^2 w\left(\mathbf{u}^1,\mathbf{u}^2\right)\mathrm{d}\mathbf{u}^1\mathrm{d}\mathbf{u}^2}, \tag{18}$$

$$I_{\mathrm{dCor}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = \begin{cases}\frac{I_{\mathrm{dCov}}\left(\mathbf{y}^1,\mathbf{y}^2\right)}{\sqrt{I_{\mathrm{dVar}}\left(\mathbf{y}^1,\mathbf{y}^1\right)I_{\mathrm{dVar}}\left(\mathbf{y}^2,\mathbf{y}^2\right)}}, & \text{if } I_{\mathrm{dVar}}\left(\mathbf{y}^1,\mathbf{y}^1\right)I_{\mathrm{dVar}}\left(\mathbf{y}^2,\mathbf{y}^2\right)>0,\\ 0, & \text{otherwise,}\end{cases}, \tag{19}$$

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

$$I_{\text{3-Lanc}}\left(\mathbf{y}^1,\mathbf{y}^2,\mathbf{y}^3\right) = \left\|\mu_{L(f)}\right\|^2_{\otimes_{i=1}^3\mathcal{H}(k_i)}, \qquad L(f) = f - f_{12}f_3 - f_{23}f_1 - f_{13}f_2 + 2f_1f_2f_3. \tag{20}$$

$$I_{\text{3-joint}}\left(\mathbf{y}^1,\mathbf{y}^2,\mathbf{y}^3\right) = \left\|\mu_{J(f)}\right\|^2_{\otimes_{i=1}^3\mathcal{H}(k_i)}, \qquad J(f) = f - f_1f_2f_3, \tag{21}$$

where $C_{\mathbf{y}^1\mathbf{y}^2}$ is the so-called cross-covariance operator, $\|\cdot\|_{\mathrm{HS}}$ is the Hilbert-Schmidt norm, $L(f)$ is the Lancaster interaction measure, μ is the mean embedding to either $\mathcal{H}(k_i)$ [for I_{HSIC}] or to $\otimes_{i=1}^3\mathcal{H}(k_i)$ [in case of $I_{\text{3-Lanc}}$ and $I_{\text{3-joint}}$], φ_m is the canonical feature map associated to kernel k_m.

- Note:

  1. $I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right)\geq 0$, $I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right)=0 \Leftrightarrow \mathbf{y}^m$-s are jointly independent.
  2. $I_{\mathrm{R},\alpha}$, $I_{\mathrm{T},\alpha}$: $\lim_{\alpha\to1}I_{\mathrm{R},\alpha}(\mathbf{y}) = \lim_{\alpha\to1}I_{\mathrm{T},\alpha}(\mathbf{y}) = I(\mathbf{y})$.
  3. I_{KCCA}: KCCA captures the maximal correlation in the $\mathcal{H}(k_i)$ feature spaces. It can be generalized to $M\geq2$ components to measure pairwise independence; this extension is available in ITE.
  4. I_{KGV}: KGV is the extension of the analytical expression of mutual information holding for Gaussian variables $(\mathbf{y})$.
  5. I_{Φ}: It is a multivariate $(M\geq2)$ extension of Hoeffding's Φ capturing the deviation from the product copula in $L^2([0,1]^d)$ sense. $h_2(d)$ is a normalizing constant ensuring that $I_{\Phi}(C)\in[0,1]$ for any copula C.

6. I_{HSIC}: It can also be extended to the $M \geq 2$ case to measure pairwise independence; the generalization is available in ITE.
7. I_{dCov}:
   - It measures independence by the difference of the joint characteristic function and the product of the marginals, in L_w^2 sense.
   - For $\alpha = 1$ the distance covariance can be rewritten in terms of pairwise distances

$$\begin{aligned} I_{\text{dCov}}\left(\mathbf{y}^1\mathbf{y}^2\right) &= \mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\mathbb{E}_{\widetilde{\mathbf{y}^1}\widetilde{\mathbf{y}^2}}\left[\left\|\mathbf{y}^1 - \widetilde{\mathbf{y}^1}\right\|_2 \left\|\mathbf{y}^2 - \widetilde{\mathbf{y}^2}\right\|_2\right] + \mathbb{E}_{\mathbf{y}^1\widetilde{\mathbf{y}^1}}\left[\left\|\mathbf{y}^1 - \widetilde{\mathbf{y}^1}\right\|_2\right]\mathbb{E}_{\mathbf{y}^2\widetilde{\mathbf{y}^2}}\left[\left\|\mathbf{y}^2 - \widetilde{\mathbf{y}^2}\right\|_2\right] \\ &\quad - 2\mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\left[\mathbb{E}_{\widetilde{\mathbf{y}^1}}\left\|\mathbf{y}^1 - \widetilde{\mathbf{y}^1}\right\|_2 \mathbb{E}_{\widetilde{\mathbf{y}^2}}\left\|\mathbf{y}^2 - \widetilde{\mathbf{y}^2}\right\|_2\right]. \end{aligned} \tag{22}$$

This form has a natural extension to semimetric spaces $[\mathbf{y}^1 \in (\mathcal{Y}_1, \rho_1),\ \mathbf{y}^2 \in (\mathcal{Y}_2, \rho_2)]$ of negative type:

$$\begin{aligned} I_{\text{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right) &= \mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\mathbb{E}_{\widetilde{\mathbf{y}^1}\widetilde{\mathbf{y}^2}}\left[\rho_1\left(\mathbf{y}^1, \widetilde{\mathbf{y}^1}\right)\rho_2\left(\mathbf{y}^2, \widetilde{\mathbf{y}^2}\right)\right] + \mathbb{E}_{\mathbf{y}^1\widetilde{\mathbf{y}^1}}\left[\rho_1\left(\mathbf{y}^1, \widetilde{\mathbf{y}^1}\right)\right]\mathbb{E}_{\mathbf{y}^2\widetilde{\mathbf{y}^2}}\left[\rho_2\left(\mathbf{y}^2, \widetilde{\mathbf{y}^2}\right)\right] \\ &\quad - 2\mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\left(\mathbb{E}_{\widetilde{\mathbf{y}^1}}\left[\rho_1\left(\mathbf{y}^1, \widetilde{\mathbf{y}^1}\right)\right]\mathbb{E}_{\widetilde{\mathbf{y}^2}}\left[\left(\mathbf{y}^2, \widetilde{\mathbf{y}^2}\right)\right]\right), \end{aligned}$$

which is proportional to HSIC (determined by kernel k):

$$\begin{aligned} I_{\text{dCov}}\left(\mathbf{y}^1, \mathbf{y}^2\right) = 2I_{\text{HSIC}}\left(\mathbf{y}^1, \mathbf{y}^2\right), \qquad k((\mathbf{u}_1, \mathbf{v}_1), (\mathbf{u}_2, \mathbf{v}_2)) &= k_1(\mathbf{u}_1, \mathbf{u}_2)k_2(\mathbf{v}_1, \mathbf{v}_2), \\ \rho_i(\mathbf{u}, \mathbf{v}) &= k_i(\mathbf{u}, \mathbf{u}) + k_i(\mathbf{v}, \mathbf{v}) - 2k_i(\mathbf{u}, \mathbf{v}). \end{aligned} \tag{23}$$

8. I_{dCor}: is the normalized variant of I_{dCov}; $I_{\text{dCor}}\left(\mathbf{y}^1, \mathbf{y}^2\right) \in [0, 1]$. It is zero iff $\mathbf{y}^1$ and $\mathbf{y}^2$ are independent.
9. $I_{\text{3-Lanc}}$: guaranteed to be zero, if f can be factorised as a product of its (possible) multidimensional marginals, i.e. $\left([\mathbf{y}^1; \mathbf{y}^2] \perp\!\!\!\perp \mathbf{y}^3\right) \vee \left([\mathbf{y}^1; \mathbf{y}^3] \perp\!\!\!\perp \mathbf{y}^2\right) \vee \left([\mathbf{y}^2; \mathbf{y}^3] \perp\!\!\!\perp \mathbf{y}^1\right) \Rightarrow L(f) = 0$. For example, '$[\mathbf{y}^1; \mathbf{y}^2] \perp\!\!\!\perp \mathbf{y}^3$' stands for $f = f_{12}f_3$.
10. $I_{\text{3-joint}}$: measures joint independence in $\otimes_{i=1}^3 \mathcal{H}(k_i)$.

3.1.3 Divergence

- Notations: $\mathbb{R}^d \ni \mathbf{y}_1 \sim f_1, \mathbb{R}^d \ni \mathbf{y}_2 \sim f_2$. $\underline{\pi_1\mathbf{y}^1 + \pi_2\mathbf{y}^2}$ is the mixture distribution obtained from $\mathbf{y}^1$ and $\mathbf{y}^2$ with π_1, π_2 weights ($\pi_1, \pi_2 > 0$, $\pi_1 + \pi_2 = 1$). $\widetilde{\mathbf{y}^1}$ and $\widetilde{\mathbf{y}^2}$ are identically distributed copies of $\mathbf{y}^1$ and $\mathbf{y}^2$. *supp*(f_i) is the support of the pdf f_i.

- Goal: Given independent, i.i.d. samples from f_1 and f_2, $\{\mathbf{y}_t^1\}_{t=1}^{T_1}$ and $\{\mathbf{y}_t^2\}_{t=1}^{T_2}$, we want to estimate the divergence of the two underlying random variables ($\mathbf{y}_1$ and $\mathbf{y}_2$).

- Definitions: The Kullback-Leibler divergence (D; also called relative entropy or I directed divergence), Rényi divergence ($D_{\text{R},\alpha}$), Tsallis divergence ($D_{\text{T},\alpha}$), Sharma-Mittal divergence ($D_{\text{SM},\alpha,\beta}$), Pearson χ^2 divergence (D_{χ^2}; also called χ^2 distance), Hellinger distance (D_{H}), L_2 divergence (D_{L_2}), (Csiszár) f-divergence (D_f; also called Csiszár-Morimoto divergence or Ali-Silvey distance), maximum mean discrepancy (D_{MMD}; MMD, also called kernel distance, current distance), energy distance (D_{EnDist}; also called N-distance), Bhattacharyya distance (D_{B}), non-symmetric Bregman distance ($D_{\text{NB},\alpha}$; also called Bregman divergence), symmetric Bregman distance ($D_{\text{SB},\alpha}$), J-distance (D_{J}; symmetrised Kullback-Leibler divergence, J divergence), K divergence (D_{K}), L divergence (D_{L}) Jensen-Shannon divergence (D_{JS}^{π}), Jensen-Rényi divergence ($D_{\text{JR},\alpha}^{\pi}$), Jensen-Tsallis divergence ($D_{\text{JT},\alpha}$) are defined as

$$D(f_1, f_2) = \int_{\mathbb{R}^d} f_1(\mathbf{u}) \log\left[\frac{f_1(\mathbf{u})}{f_2(\mathbf{u})}\right] \mathrm{d}\mathbf{u}, \tag{24}$$

$$D_{\text{R},\alpha}(f_1, f_2) = \frac{1}{\alpha - 1} \log \int_{\mathbb{R}^d} f_1^{\alpha}(\mathbf{u}) f_2^{1-\alpha}(\mathbf{u}) \mathrm{d}\mathbf{u}, \quad (\alpha \in \mathbb{R} \setminus \{1\}), \tag{25}$$

$$D_{\text{T},\alpha}(f_1, f_2) = \frac{1}{\alpha - 1}\left(\int_{\mathbb{R}^d} f_1^{\alpha}(\mathbf{u}) f_2^{1-\alpha}(\mathbf{u}) \mathrm{d}\mathbf{u} - 1\right), \quad (\alpha \in \mathbb{R} \setminus \{1\}), \tag{26}$$

$$D_{\text{SM},\alpha,\beta}(f_1, f_2) = \frac{1}{\beta - 1}\left[\left(\int_{\mathbb{R}^d} [f_1(\mathbf{u})]^{\alpha}[f_2(\mathbf{u})]^{1-\alpha} \mathrm{d}\mathbf{u}\right)^{\frac{1-\beta}{1-\alpha}} - 1\right], \quad (\alpha \neq 1, \beta \neq 1), \tag{27}$$

$$D_{\chi^2}(f_1,f_2)=\int_{supp(f_1)\cup supp(f_2)}\frac{\left[f_1(\mathbf{u})-f_2(\mathbf{u})\right]^2}{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}=\int_{supp(f_1)\cup supp(f_2)}\frac{\left[f_1(\mathbf{u})\right]^2}{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}-1, \tag{28}$$

$$D_{\mathrm{H}}(f_1,f_2)=\sqrt{\frac{1}{2}\int_{\mathbb{R}^d}\left[\sqrt{f_1(\mathbf{u})}-\sqrt{f_2(\mathbf{u})}\right]^2\mathrm{d}\mathbf{u}}=\sqrt{1-\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}}, \tag{29}$$

$$D_{\mathrm{L}_2}(f_1,f_2)=\sqrt{\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})-f_2(\mathbf{u})\right]^2\mathrm{d}\mathbf{u}}, \tag{30}$$

$$D_f(f_1,f_2)=\int_{\mathbb{R}^d}f\left[\frac{f_1(\mathbf{u})}{f_2(\mathbf{u})}\right]f_2(\mathbf{u})\mathrm{d}\mathbf{u},\quad f\text{: convex, } f(1)=0, \tag{31}$$

$$D_{\mathrm{MMD}}(f_1,f_2)=\left\|\mu_1-\mu_2\right\|_{\mathcal{H}(k)}, \tag{32}$$

$$\begin{aligned}D_{\mathrm{EnDist}}(f_1,f_2)&=2\mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\left[\rho\left(\mathbf{y}^1,\mathbf{y}^2\right)\right]-\mathbb{E}_{\mathbf{y}^1\widetilde{\mathbf{y}^1}}\left[\rho\left(\mathbf{y}^1,\widetilde{\mathbf{y}^1}\right)\right]-\mathbb{E}_{\mathbf{y}^2\widetilde{\mathbf{y}^2}}\left[\rho\left(\mathbf{y}^2,\widetilde{\mathbf{y}^2}\right)\right]\xrightarrow{\text{specifically: }\rho(\mathbf{u},\mathbf{v})=\|\mathbf{u}-\mathbf{v}\|_2^{\alpha}}\\&=2\mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\left\|\mathbf{y}^1-\mathbf{y}^2\right\|_2^{\alpha}-\mathbb{E}_{\mathbf{y}^1\widetilde{\mathbf{y}^1}}\left\|\mathbf{y}^1-\widetilde{\mathbf{y}^1}\right\|_2^{\alpha}-\mathbb{E}_{\mathbf{y}^2\widetilde{\mathbf{y}^2}}\left\|\mathbf{y}^2-\widetilde{\mathbf{y}^2}\right\|_2^{\alpha},\ \alpha\in(0,2),\end{aligned} \tag{33}$$

$$D_{\mathrm{B}}(f_1,f_2)=-\log\left(\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}\right), \tag{34}$$

$$D_{\mathrm{NB},\alpha}(f_1,f_2)=\int_{\mathbb{R}^d}\left[f_2^{\alpha}(\mathbf{u})+\frac{1}{\alpha-1}f_1^{\alpha}(\mathbf{u})-\frac{\alpha}{\alpha-1}f_1(\mathbf{u})f_2^{\alpha-1}(\mathbf{u})\right]\mathrm{d}\mathbf{u},\quad(\alpha\neq 1), \tag{35}$$

$$D_{\mathrm{SB},\alpha}(f_1,f_2)=\frac{1}{\alpha}\left[D_{\mathrm{NB},\alpha}(f_1,f_2)+D_{\mathrm{NB},\alpha}(f_2,f_1)\right],\quad(\alpha\neq 1) \tag{36}$$

$$=\frac{1}{\alpha-1}\int_{\mathbb{R}^d}f_1^{\alpha}(\mathbf{u})+f_2^{\alpha}(\mathbf{u})-f_1(\mathbf{u})f_2^{\alpha-1}(\mathbf{u})-f_2(\mathbf{u})f_1^{\alpha-1}(\mathbf{u})\mathrm{d}\mathbf{u}, \tag{37}$$

$$D_{\mathrm{J}}(f_1,f_2)=D(f_1,f_2)+D(f_2,f_1), \tag{38}$$

$$D_{\mathrm{K}}(f_1,f_2)=D\left(f_1,\frac{f_1+f_2}{2}\right), \tag{39}$$

$$D_{\mathrm{L}}(f_1,f_2)=D_{\mathrm{K}}(f_1,f_2)+D_{\mathrm{K}}(f_2,f_1), \tag{40}$$

$$D_{\mathrm{JS}}^{\pi}(f_1,f_2)=H\left(\pi_1\mathbf{y}^1+\pi_2\mathbf{y}^2\right)-\left[\pi_1H\left(\mathbf{y}^1\right)+\pi_2H\left(\mathbf{y}^2\right)\right], \tag{41}$$

$$D_{\mathrm{JS}}(f_1,f_2)=D_{\mathrm{JS}}^{\left(\frac{1}{2},\frac{1}{2}\right)}(f_1,f_2)$$

$$D_{\mathrm{JR},\alpha}^{\pi}(f_1,f_2)=H_{\mathrm{R},\alpha}\left(\pi_1\mathbf{y}^1+\pi_2\mathbf{y}^2\right)-\left[\pi_1H_{\mathrm{R},\alpha}\left(\mathbf{y}^1\right)+\pi_2H_{\mathrm{R},\alpha}\left(\mathbf{y}^2\right)\right],\quad(0<\alpha\neq 1), \tag{42}$$

$$D_{\mathrm{JR},\alpha}(f_1,f_2)=D_{\mathrm{JR},\alpha}^{\left(\frac{1}{2},\frac{1}{2}\right)}(f_1,f_2)$$

$$D_{\mathrm{JT},\alpha}(f_1,f_2)=H_{\mathrm{T},\alpha}\left(\frac{\mathbf{y}^1+\mathbf{y}^2}{2}\right)-\frac{H_{\mathrm{T},\alpha}\left(\mathbf{y}^1\right)+H_{\mathrm{T},\alpha}\left(\mathbf{y}^2\right)}{2},\quad(\alpha\neq 1), \tag{43}$$

where $\mu_m=\mathbb{E}_{\mathbf{y}_m\sim f_m}[k(\cdot,\mathbf{y}_m)]$ is the mean embedding of f_m to the RKHS $\mathcal{H}(k)$.[12]

- Note:
  - D: $D(f_1,f_2)\geq 0$. $D(f_1,f_2)=0\Leftrightarrow f_1=f_2$. It is a specific f-divergence [see (31)] with $f(t)=t\log(t)$.
  - $D_{\mathrm{R},\alpha}$, $D_{\mathrm{T},\alpha}$:
    - $\lim_{\alpha\to 1}D_{\mathrm{R},\alpha}(f_1,f_2)=\lim_{\alpha\to 1}D_{\mathrm{T},\alpha}(f_1,f_2)=D(f_1,f_2)$.
    - $\alpha<0\Rightarrow D_{\mathrm{R},\alpha}(f_1,f_2)\leq 0, D_{\mathrm{T},\alpha}(f_1,f_2)\leq 0$.
    - $\alpha=0\Rightarrow D_{\mathrm{R},\alpha}(f_1,f_2)=D_{\mathrm{T},\alpha}(f_1,f_2)=0$.
    - $\alpha>0\Rightarrow D_{\mathrm{R},\alpha}(f_1,f_2)\geq 0, D_{\mathrm{T},\alpha}(f_1,f_2)\geq 0$.
  - $D_{\mathrm{SM},\alpha,\beta}(f_1,f_2)$: $D_{\mathrm{SM},\alpha,\beta}(f_1,f_2)=0$, if and only if $f_1=f_2$.

$$D_{\mathrm{SM},\alpha,\beta}(f_1,f_2)=\frac{1}{\beta-1}\left(\left[D_{\mathrm{temp1}}(\alpha)\right]^{\frac{1-\beta}{1-\alpha}}-1\right),\qquad D_{\mathrm{temp1}}(\alpha)=\int_{\mathbb{R}^d}\left[f_1(\mathbf{u})\right]^{\alpha}\left[f_2(\mathbf{u})\right]^{1-\alpha}\mathrm{d}\mathbf{u}. \tag{44}$$

$D_{\mathrm{temp1}}(\alpha)$ is the α-divergence, or for $\alpha=\frac{1}{2}$ the Bhattacharyya coefficient (also called Bhattacharyya kernel, or Hellinger affinity; a specific case of probability product kernels, see (59)): $BC=\int_{\mathbb{R}^d}\sqrt{f_1(\mathbf{u})}\sqrt{f_2(\mathbf{u})}\mathrm{d}\mathbf{u}\in[0,1]$.

[12] We use the shorthand $k(\cdot,x)$ to denote the mapping $y\mapsto k(y,x)$.

D_{temp1} is a specific case of $D_{\text{temp2}}(a,b) = \int_{\mathbb{R}^d} [f_1(\mathbf{u})]^a [f_2(\mathbf{u})]^b f_1(\mathbf{u})\mathrm{d}\mathbf{u}$, $(a,b \in \mathbb{R})$; several divergences can be expressed by these quantities.

- D_{H}^2: is an f-divergence [(31)] with $f(t) = \frac{1}{2}(\sqrt{t}-1)^2$.
- $D_{\mathrm{L}_2}(f_1,f_2)$ is non-negative, and is zero iff $f_1 = f_2$.
- D_f: $D_f(f_1,f_2) \ge 0$ with equality iff $f_1 = f_2$.
- D_{MMD}:
    - MMD is a specific case of integral probability metrics: $D_{\mathrm{MMD}}(f_1,f_2) = \sup_{g\in\mathcal{B}}\left(\mathbb{E}_{\mathbf{y}^1\sim f_1}\left[g\left(\mathbf{y}^1\right)\right] - \mathbb{E}_{\mathbf{y}^2\sim f_2}[g\left(\mathbf{y}^2\right)]\right)$ with $\mathcal{B} := \{g : \|g\|_{\mathcal{H}(k)} \le 1\}$ being the unit ball in $\mathcal{H}(k)$.
    - MMD can be defined on topological spaces.
    - It also 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 (7) - (11)):

$$I_{\mathrm{HSIC}}\left(\mathbf{y}^1,\mathbf{y}^2\right) = D_{\mathrm{MMD}}(f, f_1f_2), \quad \left[\mathbf{y}^1;\mathbf{y}^2\right] \sim f.$$

- D_{EnDist}:
    - For $\rho(u,v) = |u-v|$, D_{EnDist} is twice the Cramer-Von Mises distance.
    - The construction holds for $(\mathcal{Z},\rho)$ semimetric spaces of negative type.
    - $D_{\mathrm{EnDist}}(f_1,f_2) \ge 0$. $D_{\mathrm{EnDist}}(f_1,f_2) = 0 \Leftrightarrow f_1 = f_2$ for strictly negative spaces (such as $\mathbb{R}^d$).
- $D_{\mathrm{SB},\alpha}$: For $\alpha = 2$, $[D_{\mathrm{L}}(f_1,f_2)]^2 = D_{\mathrm{NB},2}(f_1,f_2) = D_{\mathrm{SB},2}(f_1,f_2)$.
- D_K, D_L: They are
    - non-negative, and are zero iff $f_1 = f_2$.
    - closely related to the Jensen-Shannon divergence in case of uniform weighting, see (45).
- D_{JS}^{π}:
    - $0 \le D_{\mathrm{JS}}^{\pi}(f_1,f_2) \le \log(2)$, $D_{\mathrm{JS}}^{\pi}(f_1,f_2) = 0 \Leftrightarrow f_1 = f_2$.
    - Specifically, for $\pi_1 = \pi_2 = \frac{1}{2}$ we obtain

$$\begin{aligned} D_{\mathrm{JS}}(f_1,f_2) &= D_{\mathrm{JS}}^{\left(\frac{1}{2},\frac{1}{2}\right)}(f_1,f_2) \\ &= H\left(\frac{\mathbf{y}^1+\mathbf{y}^2}{2}\right) - \frac{H\left(\mathbf{y}^1\right)+H\left(\mathbf{y}^2\right)}{2} = \frac{1}{2}\left[D\left(f_1,\frac{f_1+f_2}{2}\right) + D\left(f_2,\frac{f_1+f_2}{2}\right)\right]. \end{aligned} \tag{45}$$

- $D_{\mathrm{JT},\alpha}$: $\lim_{\alpha\to1} D_{\mathrm{JT},\alpha}(f_1,f_2) = D_{\mathrm{JS}}(f_1,f_2)$.

3.1.4 Association Measure

- Notations: $\mathbf{y} = \left[\mathbf{y}^1;\ldots;\mathbf{y}^M\right] \in \mathbb{R}^d$ $(\mathbf{y}^m \in \mathbb{R}^{d_m}$, $d = \sum_{m=1}^M d_m)$. F: cdf (cumulative density function) of $\mathbf{y} = \left[y^1;\ldots;y^d\right]$, F_i: cdf of y^i. C: copula of $\mathbf{y}$, i.e. $F(\mathbf{y}) = C\left(F_1\left(y^1\right),\ldots,F_d\left(y^d\right)\right)$, in other words $C(\mathbf{u}) = \mathbb{P}(\mathbf{U} \le \mathbf{u})$ where $\mathbf{U} = \left[F_1\left(y^1\right);\ldots;F_d\left(y^d\right)\right] \in [0,1]^d$. Bar stands for the survival function of its argument (it is *not* a copula in general): $\bar{C}(\mathbf{u}) := \mathbb{P}(\mathbf{U} > \mathbf{u})$. C_{kl}: bivariate marginal copula of y_{kl}, $\Pi(u_1,\ldots,u_d) = \prod_{i=1}^d u_i$ (product copula), $M(\mathbf{u}) = \min_{i=1,\ldots,d} u_i$ (comonotonicity copula)[13]. The name of M originates from the fact that for any C copula

$$W(\mathbf{u}) := \max(u_1+\ldots+u_d-d+1,0) \le C(\mathbf{u}) \le M(\mathbf{u}), \quad \forall\mathbf{u}\in[0,1]^d. \tag{46}$$

The well-known Spearman's ρ (also called Spearman's rank correlation coefficient or grade correlation coefficient) is

$$\begin{aligned} A_\rho\left(y^1,y^2\right) &= corr\left(F_1\left(y^1\right),F_2\left(y^2\right)\right) \\ &= A_\rho(C) = \frac{\int_{[0,1]^2} u_1u_2\mathrm{d}C(\mathbf{u}) - \left(\frac{1}{2}\right)^2}{\frac{1}{12}} = 12\int_{[0,1]^2} C(\mathbf{u})\mathrm{d}\mathbf{u} - 3 = \frac{\int_{[0,1]^2} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^2}\Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^2} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^2}\Pi(\mathbf{u})\mathrm{d}\mathbf{u}}, \end{aligned}$$

where $\int_{[0,1]^2} M(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{1}{3}$, $\int_{[0,1]^2}\Pi(\mathbf{u})\mathrm{d}\mathbf{u} = \frac{1}{4}$. A_ρ can be viewed as the normalized average difference of the copula of $\mathbf{y}$ (C) and the independence copula (Π).

[13] Notice that in this section 'M' stands for the comontonicity copula with argument $\mathbf{u}$ [see $M(\mathbf{u})$] and also for the number of subspaces in $\mathbf{y} = \left[\mathbf{y}^1;\ldots;\mathbf{y}^M\right]$ as a subscript.

- Goal: Our aim is to estimate the association of the d_m-dimensional components ($\mathbf{y}^m$) of the random variable $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right] \in \mathbb{R}^d$ from which we have i.i.d. samples $\{\mathbf{y}_t\}_{t=1}^T$ ($d = \sum_{m=1}^M d_m$, $\mathbf{y}^m \in \mathbb{R}^{d_m}$).

- Definitions: The Spearman's ρ multivariate-1 (A_{ρ_1}), Spearman's ρ multivariate-2 (A_{ρ_2}), Spearman's ρ multivariate-3 (A_{ρ_3}; average of A_{ρ_1} and A_{ρ_2}), Spearman's ρ multivariate-4 (A_{ρ_4}; average pairwise Spearman's ρ), multivariate extension of Blomqvist's β (A_β; medial correlation coefficient), multivariate conditional version of Spearman's ρ (lower tail: $A_{\rho_{\mathrm{lt}}}$, upper tail: $A_{\rho_{\mathrm{ut}}}$), lower and upper tail dependencies via conditional Spearman's ρ ($A_{\rho_{\mathrm{L}}}$, $A_{\rho_{\mathrm{U}}}$) are defined as

$$A_{\rho_1}\left(y^1, \ldots, y^d\right) = A_{\rho_1}(C) = \frac{\int_{[0,1]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}} = h_\rho(d)\left[2^d \int_{[0,1]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - 1\right], \tag{47}$$

$$h_\rho(d) = \frac{d+1}{2^d - (d+1)}, \tag{48}$$

$$A_{\rho_2}\left(y^1, \ldots, y^d\right) = A_{\rho_2}(C) = \frac{\int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}C(\mathbf{u}) - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,1]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}} = h_\rho(d)\left[2^d \int_{[0,1]^d} \Pi(\mathbf{u})\mathrm{d}C(\mathbf{u}) - 1\right], \tag{49}$$

$$A_{\rho_3}\left(y^1, \ldots, y^d\right) = A_{\rho_3}(C) = \frac{A_{\rho_1}\left(y^1, \ldots, y^d\right) + A_{\rho_2}\left(y^1, \ldots, y^d\right)}{2}, \tag{50}$$

$$A_{\rho_4}\left(y^1, \ldots, y^d\right) = A_{\rho_4}(C) = h_\rho(2)\left[2^2 \binom{d}{2}^{-1} \sum_{k,l=1;k<l}^{d} \int_{[0,1]^2} C_{kl}(u,v)\mathrm{d}u\mathrm{d}v - 1\right] = \binom{d}{2}^{-1} \sum_{k,l=1;k<l}^{d} A_\rho\left(y^k, y^l\right), \tag{51}$$

$$A_\beta\left(y^1, \ldots, y^d\right) = A_\beta(C) = \frac{C\left(\mathbf{1/2}\right) - \Pi\left(\mathbf{1/2}\right) + \bar{C}\left(\mathbf{1/2}\right) - \bar{\Pi}\left(\mathbf{1/2}\right)}{M\left(\mathbf{1/2}\right) - \Pi\left(\mathbf{1/2}\right) + \bar{M}\left(\mathbf{1/2}\right) - \bar{\Pi}\left(\mathbf{1/2}\right)} = h_\beta(d)\left[C(\mathbf{1/2}) + \bar{C}(\mathbf{1/2}) - 2^{1-d}\right], \tag{52}$$

$$h_\beta(d) = \frac{2^{d-1}}{2^{d-1} - 1},$$

$$A_{\rho_{\mathrm{lt}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\mathrm{lt}}}(C) = \frac{\int_{[0,p]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,p]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[0,p]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[0,p]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}} = \frac{\int_{[0,p]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \left(\frac{p^2}{2}\right)^d}{\frac{p^{d+1}}{d+1} - \left(\frac{p^2}{2}\right)^d}, \tag{53}$$

$$A_{\rho_{\mathrm{ut}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\mathrm{ut}}}(C) = \frac{\int_{[1-p,1]^d} C(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[1-p,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}{\int_{[1-p,1]^d} M(\mathbf{u})\mathrm{d}\mathbf{u} - \int_{[1-p,1]^d} \Pi(\mathbf{u})\mathrm{d}\mathbf{u}}, \tag{54}$$

$$A_{\rho_{\mathrm{L}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\mathrm{L}}}(C) = \lim_{p \to 0, p > 0} A_{\rho_{\mathrm{lt}}}(C) = \lim_{p \to 0, p > 0} \frac{d+1}{p^{d+1}} \int_{[0,p]^d} C(\mathbf{u})\mathrm{d}\mathbf{u}, \tag{55}$$

$$A_{\rho_{\mathrm{U}}}\left(y^1, \ldots, y^d\right) = A_{\rho_{\mathrm{U}}}(C) = \lim_{p \to 0, p > 0} A_{\rho_{\mathrm{ut}}}(C), \tag{56}$$

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

- Note:
  - $A_{\rho_1}, A_{\rho_2}, A_{\rho_3}, A_{\rho_4}$: They are generalizations of Spearman's ρ, in other words $A_\rho = A_{\rho_1} = A_{\rho_2} = A_{\rho_3}$ for $d = 2$.
  - A_{ρ_3}, A_{ρ_4}: These quantities are multivariate measures of concordance (see Def. 3 in Section C).
  - $A_{\rho_1}, A_{\rho_2}, A_\beta$: They satisfy all the axioms of multivariate measure of concordance except for Duality.
  - $A_{\rho_{\mathrm{lt}}}, A_{\rho_{\mathrm{ut}}}$: They belong to the following class of association measures parameterized by a function g

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

  - $A_{\rho_{\mathrm{lt}}}$:
    - Here $g(\mathbf{u}) = \mathbb{I}_{[0,p]^d}(\mathbf{u})$, where $0 < p \leq 1$ and $\mathbb{I}$ is 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. For $p = 1$, $A_{\rho_{\mathrm{lt}}} = A_{\rho_1}$.

    - It preserves the concordance ordering [see (78)], i.e., $C_1 \prec C_2 \Rightarrow A_{\rho_{lt}}(C_1) \leq A_{\rho_{lt}}(C_2)$, for $\forall p \in (0,1]$. Thus, from $C \prec M$ [see (46)] one obtains that $A_{\rho_{lt}} \leq 1$.
  - $A_{\rho_{\mathrm{ut}}}$: In this case $g(\mathbf{u}) = \mathbb{I}_{[1-p,1]^d}(\mathbf{u})$, where $0 < p \leq 1$; in other words the weighting is put on the upper tail.

3.1.5 Cross Quantity

- Notation: $\mathbb{R}^d \ni \mathbf{y}^1 \sim f_1$, $\mathbb{R}^d \ni \mathbf{y}^2 \sim f_2$.
- Goal: We want to estimate 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.
- Definition: The cross-entropy (C_{CE}) is

$$C_{\mathrm{CE}}(f_1, f_2) = -\int_{\mathbb{R}^d} f_1\left(\mathbf{u}\right) \log\left[f_2\left(\mathbf{u}\right)\right] \mathrm{d}\mathbf{u}. \tag{57}$$

3.1.6 Kernel on Distributions

- Notation: $\mathbb{R}^d \ni \mathbf{y}^1 \sim f_1$, $\mathbb{R}^d \ni \mathbf{y}^2 \sim f_2$.
- Goal: Our aim is to estimate the value of a kernel $[K(f_1, f_2)]$ given independent, i.i.d. samples from $\mathbf{y}^1$ and $\mathbf{y}^2$, $\{\mathbf{y}_t^1\}_{t=1}^{T_1}$ and $\{\mathbf{y}_t^2\}_{t=1}^{T_2}$.
- Definitions: the expected kernel (K_{exp}; also called summation kernel, mean map kernel, set kernel, multi-instance kernel, ensemble kernel; a specific convolution kernel), probability product kernel ($K_{\mathrm{PP},\rho}$), Jensen-Shannon kernel (K_{JS}), Jensen-Tsallis kernel ($K_{\mathrm{JT},\alpha}$), exponentiated Jensen-Shannon kernel ($K_{\mathrm{EJS},u}$), exponentiated Jensen-Rényi kernels ($K_{\mathrm{EJR1},u,\alpha}$, $K_{\mathrm{EJR2},u,\alpha}$), exponentiated Jensen-Tsallis kernels ($K_{\mathrm{EJT1},u,\alpha}$, $K_{\mathrm{EJT2},u,\alpha}$) are defined as

$$K_{\mathrm{exp}}\left(f_1, f_2\right) = \left\langle \mu_1, \mu_2 \right\rangle_{\mathcal{H}(k)} = \mathbb{E}_{\mathbf{y}^1\mathbf{y}^2}\left[k\left(\mathbf{y}^1, \mathbf{y}^2\right)\right], \qquad \mu_i = \mathbb{E}_{\mathbf{y}^i \sim f_i}\left[k(\cdot, \mathbf{y}^i)\right], \tag{58}$$

$$K_{\mathrm{PP},\rho}\left(f_1, f_2\right) = \int_{\mathbb{R}^d} \left[f_1(\mathbf{u})\right]^\rho \left[f_2(\mathbf{u})\right]^\rho \mathrm{d}\mathbf{u}, \quad (\rho > 0), \tag{59}$$

$$K_{\mathrm{JS}}\left(f_1, f_2\right) = \log(2) - D_{\mathrm{JS}}(f_1, f_2), \tag{60}$$

$$K_{\mathrm{JT},\alpha}\left(f_1, f_2\right) = \log_\alpha(2) - T_\alpha(f_1, f_2), \quad (\alpha \in (0,2]\backslash\{1\}), \qquad \log_\alpha(x) = \frac{x^{1-\alpha} - 1}{1-\alpha}, \tag{61}$$

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

$$K_{\mathrm{EJS},u}\left(f_1, f_2\right) = e^{-uD_{\mathrm{JS}}(f_1, f_2)}, \quad (u > 0), \tag{62}$$

$$K_{\mathrm{EJR1},u,\alpha}\left(f_1, f_2\right) = e^{-uH_{\mathrm{R},\alpha}\left(\frac{\mathbf{y}^1+\mathbf{y}^2}{2}\right)}, \quad (u > 0,\ \alpha \in (0,1)), \tag{63}$$

$$K_{\mathrm{EJR2},u,\alpha}\left(f_1, f_2\right) = e^{-uD_{\mathrm{JR},\alpha}(f_1, f_2)}, \quad (u > 0,\ \alpha \in (0,1)) \qquad D_{\mathrm{JR},\alpha}\left(f_1, f_2\right) = D_{\mathrm{JR},\alpha}^{\left(\frac{1}{2},\frac{1}{2}\right)}\left(f_1, f_2\right), \tag{64}$$

$$K_{\mathrm{EJT1},u,\alpha}\left(f_1, f_2\right) = e^{-uH_{\mathrm{T},\alpha}\left(\frac{\mathbf{y}^1+\mathbf{y}^2}{2}\right)}, \quad (u > 0,\ \alpha \in (0,2]\backslash\{1\}), \tag{65}$$

$$K_{\mathrm{EJT2},u,\alpha}\left(f_1, f_2\right) = e^{-uD_{\mathrm{JT},\alpha}(f_1, f_2)}, \quad (u > 0,\ \alpha \in (0,2]\backslash\{1\}). \tag{66}$$

- Note:
  - K_{exp}: it
    - generates MMD, $\left[D_{\mathrm{MMD}}(f_1, f_2)\right]^2 = K_{\mathrm{exp}}\left(f_1, f_1\right) - 2K_{\mathrm{exp}}\left(f_1, f_2\right) + K_{\mathrm{exp}}\left(f_1, f_2\right)$.
    - can be defined slightly more generally, on topological spaces.
  - $K_{\mathrm{PP},\rho}$: For $\rho = \frac{1}{2}$ we get back the Bhattacharyya kernel (K_{B}; also known as Bhattacharyya coefficient, or Hellinger affinity) K_{B} is closely related to the Hellinger distance [see (29)]:

$$\left[D_{\mathrm{H}}(f_1, f_2)\right]^2 = \frac{1}{2}\left[K_{\mathrm{B}}\left(f_1, f_1\right) - 2K_{\mathrm{B}}\left(f_1, f_2\right) + K_{\mathrm{B}}\left(f_2, f_2\right)\right] = \frac{1}{2}\left[2 - 2K_{\mathrm{B}}\left(f_1, f_2\right)\right] = 1 - K_{\mathrm{B}}\left(f_1, f_2\right). \tag{67}$$

  - $K_{\text{JT},\alpha}$: $\lim_{\alpha\to 1} K_{\text{JT},\alpha}(f_1,f_2) = K_{\text{JS}}(f_1,f_2)$.
  - $K_{\text{EJR2},u,\alpha}$: $\lim_{\alpha\to 1} K_{\text{EJR2},u,\alpha}(f_1,f_2) = K_{\text{EJS},u}(f_1,f_2)$.
  - $K_{\text{EJT2},u,\alpha}$: $\lim_{\alpha\to 1} K_{\text{EJT2},u,\alpha}(f_1,f_2) = K_{\text{EJS},u}(f_1,f_2)$.

3.2 Conditional Quantities

The toolbox supports the estimation of conditional quantities defined in Section 3.2.1 (entropy) and Section 3.2.2 (mutual information).

3.2.1 Entropy

- Notation: $\mathbf{y} = \left[\mathbf{y}^1;\mathbf{y}^2\right]$, $\mathbf{y}^m \in \mathbb{R}^{d_m}$.
- Goal: Assume we have $\left\{\left(\mathbf{y}_t^1,\mathbf{y}_t^2\right)\right\}_{t=1}^T$ samples; we want to estimate the conditional entropy of $\mathbf{y}^1$ given $\mathbf{y}^2$.
- Definition: The conditional Shannon entropy $[H(\cdot|\cdot)]$ is defined as

$$H\left(\mathbf{y}^1|\mathbf{y}^2\right) = \mathbb{E}_{\mathbf{y}^2}\left[H\left(\mathbf{y}^1|\mathbf{y}^2\right)\right] = H\left(\left[\mathbf{y}^1;\mathbf{y}^2\right]\right) - H\left(\mathbf{y}^2\right). \tag{68}$$

3.2.2 Mutual Information

- Notation: $\mathbf{y} = \left[\mathbf{y}^1;\ldots;\mathbf{y}^M;\mathbf{y}^{M+1}\right] \in \mathbb{R}^d$, where $\mathbf{y}^m \in \mathbb{R}^{d_m}$ and $d = \sum_{m=1}^{M+1} d_m$.
- Goal: Assume we have access to the samples $\left\{\left(\mathbf{y}_t^1,\ldots,\mathbf{y}_t^M,\mathbf{y}_t^{M+1}\right)\right\}_{t=1}^T$, our aim is to estimate the mutual information of $\mathbf{y}^1,\ldots,\mathbf{y}^M$ given $\mathbf{y}^{M+1}$.
- Definition: The conditional Shannon mutual information $[I(\cdot|\cdot)]$ is

$$\begin{aligned} I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M|\mathbf{y}^{M+1}\right) &= \mathbb{E}_{\mathbf{y}^{M+1}}\left[I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M|\mathbf{y}^{M+1}\right)\right] \\ &= -H\left(\left[\mathbf{y}^1;\ldots;\mathbf{y}^{M+1}\right]\right) + \sum_{m=1}^M H\left(\left[\mathbf{y}^m;\mathbf{y}^{M+1}\right]\right) - (M-1)H\left(\mathbf{y}^{M+1}\right). \end{aligned} \tag{69}$$

3.3 Estimators

The available estimators in ITE are listed in

1. **unconditional quantities**: Table 1 (entropy), Table 2 (mutual information), Table 3 (divergence), Table 4 (association measures), Table 5 (cross quantity), Table 6 (kernel on distributions).
2. **conditional quantities**: Table 7 (entropy), Table 8 (mutual information).

Demos of the estimators are enlisted in Table 9.

- Certain association measures and mutual information estimators require one-dimensional subspaces ($\forall d_m = 1$). For these estimators the toolbox provides a simplified calling syntax (without specifying $\{d_m\}_{m=1}^M$):

```
>>> a_or_i    = co.estimation(y,ds) # long (traditional) syntax
>>> a_or_i_v2 = co.estimation(y)    # short syntax; gives the same result as the previous line
```

- We use the TypeXName naming convention.
  - For example, BHShannon_KnnK = B + H + Shannon_KnnK. B means base estimator, H is for entropy, Shannon_KnnK is about the estimated quantity/technique (Shannon entropy, with k-nearest neighbors).
  - Generally, Type ∈ {B,M}, X ∈ {H, I, D, A, C, K, condH, condI}, where B-base, M-meta; H-entropy, I-mutual information, D-divergence, A-association measure, C-cross quantity, K-kernel on distributions, condH-conditional entropy, condI-conditional mutual information.

Estimated quantity	Principle	d	Cost name
Shannon entropy (H)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BHShannon_KnnK
Shannon entropy (H)	approximate slope of the inverse distribution function	$= 1$	BHShannon_SpacingV
Shannon entropy (H)	maximum entropy distribution, function set1, plug-in	$= 1$	BHShannon_MaxEnt1
Shannon entropy (H)	maximum entropy distribution, function set2, plug-in	$= 1$	BHShannon_MaxEnt2
Rényi entropy ($H_{\mathrm{R},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BHRenyi_KnnK
Rényi entropy ($H_{\mathrm{R},\alpha}$)	generalized nearest neighbors ($S \subseteq \{1,\ldots,k\}$)	≥ 1	BHRenyi_KnnS
Tsallis entropy ($H_{\mathrm{T},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BHTsallis_KnnK
Sharma-Mittal entropy ($H_{\mathrm{SM},\alpha,\beta}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BHSharmaMittal_KnnK
Φ-entropy ($H_{\Phi,w}$)	sample spacing	$= 1$	BHPhi_Spacing
Shannon entropy (H)	-KL divergence from the normal distribution: (70)	≥ 1	MHShannon_DKLN
Shannon entropy (H)	-KL divergence from the uniform distribution: (71)	≥ 1	MHShannon_DKLU
Tsallis entropy ($H_{\mathrm{T},\alpha}$)	function of the Rényi entropy: (3)	≥ 1	MHTsallis_HR

Table 1: Entropy estimators. Third column: dimension (d) constraint. Top: base methods, bottom: meta estimators.

Estimated quantity	Principle	d_m	M	Cost name
kernel canonical correlation (I_{KCCA})	sup correlation over RKHSs	≥ 1	≥ 2	BIKGV
kernel generalized variance (I_{KGV})	Gaussian mutual information of the features	≥ 1	≥ 2	BIKCCA
Hoeffding's Φ (I_Φ), multivariate	L^2 distance of the joint- and the product copula	$= 1$	≥ 2	BIHoeffding
Hilbert-Schmidt indep. criterion (I_{HSIC})	HS norm of the cross-covariance operator	≥ 1	≥ 2	BIHSIC_IChol
distance covariance (I_{dCov})	pairwise distances	≥ 1	$= 2$	BIDistCov
distance correlation (I_{dCor})	pairwise distances	≥ 1	$= 2$	BIDistCorr
Lancaster 3-variable interaction ($I_{\text{3-Lanc}}$)	embedding of the Lancaster interaction measure	≥ 1	$= 3$	BI3WayJoint
3-variable joint independence ($I_{\text{3-joint}}$)	embedding of the 'joint - product of marginals'	≥ 1	$= 3$	BI3WayLancaster
(Shannon) mutual information (I)	KL-divergence of joint & product of marginals: (7)	≥ 1	≥ 2	MIShannon_DKL
(Shannon) mutual information (I)	entropy sum of components minus joint entropy: (72)	≥ 1	≥ 2	MIShannon_HS
Rényi mutual information ($I_{\mathrm{R},\alpha}$)	Rényi divergence of joint & product of marginals: (8)	≥ 1	≥ 2	MIRenyi_DR
Rényi mutual information ($I_{\mathrm{R},\alpha}$)	minus the Rényi entropy of the joint copula: (73)	$= 1$	≥ 2	MIRenyi_HR
Tsallis mutual information ($I_{\mathrm{T},\alpha}$)	Tsallis divergence of joint & product of marginals: (9)	≥ 1	≥ 2	MITsallis_DT
χ^2 mutual information (I_{χ^2})	χ^2 divergence of joint & product of marginals: (10)	≥ 1	≥ 2	MIChi2_DChi2
L_2 mutual information (I_{L_2})	L_2-divergence of joint & product of marginals: (11)	≥ 1	≥ 2	MIL2_DL2
copula-based kernel dependency (I_{c})	MMD div. of the joint copula & uniform distr.: (12)	$= 1$	≥ 2	MIMMD_CopulaDMMD
distance covariance (I_{dCov})	pairwise distances, equivalence to HSIC: (23)	≥ 1	$= 2$	MIDistCov_HSIC

Table 2: Mutual information estimators. Third column: dimension constraint (d_m; $\mathbf{y}^m \in \mathbb{R}^{d_m}$). Fourth column: constraint for the number of components (M; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$). Top: base methods, bottom: meta estimators.

Estimated quantity	Principle	d	Cost name
Kullback-Leibler divergence (D)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDKL_KnnK
Kullback-Leibler divergence (D)	k-nearest neighbors ($S_i = \{k_i(T_i)\}$)	≥ 1	BDKL_KnnKiTi
Rényi divergence ($D_{\mathrm{R},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDRenyi_KnnK
Tsallis divergence ($D_{\mathrm{T},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDTsallis_KnnK
Sharma-Mittal divergence ($D_{\mathrm{SM},\alpha,\beta}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDSharmaMittal_KnnK
Pearson χ^2 divergence (D_{χ^2})	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDChi2_KnnK
Hellinger distance (D_{H})	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDHellinger_KnnK
L_2 divergence (D_{L_2})	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDL2_KnnK
maximum mean discrepancy (D_{MMD})	U-statistic, unbiased	≥ 1	BDMMD_UStat
maximum mean discrepancy (D_{MMD})	V-statistic, biased	≥ 1	BDMMD_VStat
maximum mean discrepancy (D_{MMD})	U-statistic, incomplete Cholesky decomposition	≥ 1	BDMMD_UStat_IChol
maximum mean discrepancy (D_{MMD})	V-statistic, incomplete Cholesky decomposition	≥ 1	BDMMD_VStat_IChol
maximum mean discrepancy (D_{MMD})	online	≥ 1	BDMMD_Online
energy distance (D_{EnDist})	pairwise distances	≥ 1	BDEnergyDist
Bhattacharyya distance (D_{B})	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDBhattacharyya_KnnK
Bregman distance ($D_{\mathrm{NB},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDBregman_KnnK
symmetric Bregman distance ($D_{\mathrm{SB},\alpha}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BDSymBregman_KnnK
Kullback-Leibler divergence (D)	difference of cross-entropy and entropy: (74)	≥ 1	MDKL_HSCE
f-divergence (D_{f})	second-order Taylor expansion, χ^2 divergence: (75)	≥ 1	MDf_DChi2
maximum mean discrepancy (D_{MMD})	block-average of U-statistic based MMDs	≥ 1	MDBlockMMD
energy distance (D_{EnDist})	pairwise distances, equivalence to MMD: (76)	≥ 1	MDEnergyDist_DMMD
symmetric Bregman distance ($D_{\mathrm{SB},\alpha}$)	symmetrised Bregman distance: (36)	≥ 1	MDSymBregman_DB
J-distance (D_{J})	symmetrised Kullback-Leibler divergence: (38)	≥ 1	MDJDist_DKL
K divergence (D_{K})	smoothed Kullback-Leibler divergence: (39)	≥ 1	MDK_DKL
L divergence (D_{L})	symmetrised K divergence: (40)	≥ 1	MDL_DKL
Jensen-Shannon divergence (D_{JS}^{π})	smoothed (π), defined via the Shannon entropy: (41)	≥ 1	MDJS_HS
Jensen-Rényi divergence ($D_{\mathrm{JR},\alpha}^{\pi}$)	smoothed (π), defined via the Rényi entropy: (42)	≥ 1	MDJR_HR
Jensen-Tsallis divergence ($D_{\mathrm{JT},\alpha}$)	smoothed, defined via the Tsallis entropy: (43)	≥ 1	MDJT_HT

Table 3: Divergence estimators. Third column: dimension (d) constraint. Top: base methods, bottom: meta estimators.

Estimated quantity	Principle	d_m	M	Cost name
Spearman's ρ: multivariate1 (A_{ρ_1})	empirical copula, explicit formula	$= 1$	≥ 2	BASpearman1
Spearman's ρ: multivariate2 (A_{ρ_2})	empirical copula, explicit formula	$= 1$	≥ 2	BASpearman2
Spearman's ρ: multivariate3 (A_{ρ_3})	average of ρ_1 and ρ_2	$= 1$	≥ 2	BASpearman3
Spearman's ρ: multivariate4 (A_{ρ_4})	average pairwise Spearman's ρ	$= 1$	≥ 2	BASpearman4
Blomqvist's β (A_{β})	empirical copula, explicit formula	$= 1$	≥ 2	BABlomqvist
conditional Spearman's ρ, lower tail ($A_{\rho_{\mathrm{lt}}}$)	empirical copula, explicit formula	$= 1$	≥ 2	BASpearmanCondLT
conditional Spearman's ρ, upper tail ($A_{\rho_{\mathrm{ut}}}$)	empirical copula, explicit formula	$= 1$	≥ 2	BASpearmanCondUT
lower tail dep. via conditional Spearman's ρ ($A_{\rho_{\mathrm{L}}}$)	limit of $A_{\rho_{\mathrm{lt}}}$: (55)	$= 1$	≥ 2	MASpearmanLT
upper tail dep. via conditional Spearman's ρ ($A_{\rho_{\mathrm{U}}}$)	limit of $A_{\rho_{\mathrm{ut}}}$: (56)	$= 1$	≥ 2	MASpearmanUT

Table 4: Association measure estimators. Third column: dimension constraint (d_m; $\mathbf{y}^m \in \mathbb{R}^{d_m}$). Fourth column: constraint for the number of components (M; $\mathbf{y} = \left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]$). Top: base methods, bottom: meta estimators.

Estimated quantity	Principle	d	Cost name
cross-entropy (C_{CE})	k-nearest neighbors ($S = \{k\}$)	≥ 1	BCCE_KnnK

Table 5: Cross quantity estimators. Third column: dimension (d) constraint.

Estimated quantity	Principle	d	Cost name
expected kernel (K_{exp})	mean of pairwise kernel values	≥ 1	BKExpected
probability product kernel ($K_{\mathrm{PP},\rho}$)	k-nearest neighbors ($S = \{k\}$)	≥ 1	BKProbProd_KnnK
Jensen-Shannon kernel (K_{JS})	function of the Jensen-Shannon divergence: (60)	≥ 1	MKJS_DJS
Jensen-Tsallis kernel ($K_{\mathrm{JT},\alpha}$)	function of the Tsallis entropy: (61)	≥ 1	MKJT_HT
exponentiated Jensen-Shannon kernel ($K_{\mathrm{EJS},u}$)	function of the Jensen-Shannon divergence: (62)	≥ 1	MKExpJS_DJS
exponentiated Jensen-Rényi kernel-1 ($K_{\mathrm{EJR1},u,\alpha}$)	function of the Rényi entropy: (63)	≥ 1	MKExpJR1_HR
exponentiated Jensen-Rényi kernel-2 ($K_{\mathrm{EJR2},u,\alpha}$)	function of the Jensen-Rényi divergence: (64)	≥ 1	MKExpJR2_DJR
exponentiated Jensen-Tsallis kernel-1 ($K_{\mathrm{EJT1},u,\alpha}$)	function of the Tsallis entropy: (65)	≥ 1	MKExpJT1_HT
exponentiated Jensen-Tsallis kernel-2 ($K_{\mathrm{EJT2},u,\alpha}$)	function of the Jensen-Tsallis divergence: (66)	≥ 1	MKExpJT2_DJT

Table 6: Estimators of kernels on distributions. Third column: dimension (d) constraint. Top: base methods, bottom: meta estimators.

Estimated quantity	Principle	d_m	Cost name
conditional Shannon entropy [$H(\cdot\vert\cdot)$]	reduction to Shannon entropy	≥ 1	BcondHShannon_HShannon

Table 7: Conditional entropy estimators. Third column: dimension (d_m) constraint.

- Meta estimators: The rules not yet covered for the meta estimators are as follows.
  - Notation: $\mathbf{y} \sim f$; f_U: uniform density on $[0,1]^d$; $N(\mathbf{m},\mathbf{\Sigma})$: normal distribution with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$. $\mathrm{cov}(\mathbf{y})$ denotes the covariance of $\mathbf{y}$.
  - Rules:

$$H(\mathbf{y}) = H(\mathbf{y}_G) - D(f, f_G) \qquad \mathbf{y}_G \sim f_G = N(\mathbb{E}(\mathbf{y}), \mathrm{cov}(\mathbf{y})), \tag{70}$$

$$H(\mathbf{y}) = -D(f, f_U) \qquad \mathbf{y} \in [0,1]^d \text{ (if } \mathbf{y} \in [\mathbf{a},\mathbf{b}] = \times_{i=1}^{d} [a_i, b_i], \tag{71}$$

$$\text{it is linearly transformed to } [0,1]^d),$$

$$I\left(\mathbf{y}^1, \ldots, \mathbf{y}^M\right) = \sum_{m=1}^{M} H\left(\mathbf{y}^m\right) - H\left(\left[\mathbf{y}^1; \ldots; \mathbf{y}^M\right]\right), \tag{72}$$

$$I_{\mathrm{R},\alpha}\left(y^1, \ldots, y^M\right) = -H_{\mathrm{R},\alpha}(\mathbf{z}), \qquad \mathbf{z} = \left[F_1\left(y^1\right); \ldots; F_M\left(y^M\right)\right] \in \mathbb{R}^M, \tag{73}$$

$$D(f_1, f_2) = C_{\mathrm{CE}}(f_1, f_2) - H(f_1), \tag{74}$$

$$D_f(f_1, f_2) \approx \frac{f''(1)}{2} D_{\chi^2}(f_1, f_2), \tag{75}$$

$$D_{\mathrm{EnDist}}(f_1, f_2) = 2\left[D_{\mathrm{MMD}}(f_1, f_2)\right]^2, \qquad \rho(\mathbf{u},\mathbf{v}) = k(\mathbf{u},\mathbf{u}) + k(\mathbf{v},\mathbf{v}) - 2k(\mathbf{u},\mathbf{v}), \tag{76}$$

  where D_{EnDist} is determined by ρ, D_{MMD} by k.

- Base estimators: Their equations can be looked up by using Section B.

Estimated quantity	Principle	d_m	M	Cost name
conditional Shannon mutual information [$I(\cdot\vert\cdot)$]	reduction to Shannon entropy	≥ 1	≥ 2	BcondIShannon_HShannon

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

Estimated quantity	ite/demos/
Shannon entropy (H)	demo_h_shannon.py
Rényi entropy ($H_{\mathrm{R},\alpha}$)	demo_h_renyi.py
Tsallis entropy ($H_{\mathrm{T},\alpha}$)	demo_h_tsallis.py
Sharma-Mittal entropy ($H_{\mathrm{SM},\alpha,\beta}$)	demo_h_sharma_mittal.py
Φ-entropy ($H_{\Phi,w}$)	demo_h_phi.py
(Shannon) mutual information (I)	demo_i_shannon.py
Rényi mutual information ($I_{\mathrm{R},\alpha}$)	demo_i_renyi.py
Kullback-Leibler divergence (D)	demo_d_kullback_leibler.py
Rényi divergence ($D_{\mathrm{R},\alpha}$)	demo_d_renyi.py
Tsallis divergence ($D_{\mathrm{T},\alpha}$)	demo_d_tsallis.py
Sharma-Mittal divergence ($D_{\mathrm{SM},\alpha,\beta}$)	demo_d_sharma_mittal.py
Pearson χ^2 divergence (D_{χ^2})	demo_d_chi_square.py
Hellinger distance (D_{H})	demo_d_hellinger.py
L_2 divergence (D_{L_2})	demo_d_l2.py
Maximum mean discrepancy (D_{MMD})	demo_d_mmd.py
Bregman distance ($D_{\mathrm{NB},\alpha}$)	demo_d_bregman.py
Jensen-Rényi divergence ($D^{\pi}_{\mathrm{JR},\alpha}$)	demo_d_jensen_renyi.py
cross-entropy (C_{CE})	demo_c_cross_entropy.py
expected kernel (K_{exp})	demo_k_expected.py
probability product kernel ($K_{\mathrm{PP},\rho}$)	demo_k_prob_product.py
exponentiated Jensen-Rényi kernel-1 ($K_{\mathrm{EJR1},u,\alpha}$)	demo_k_ejr1.py
exponentiated Jensen-Rényi kernel-2 ($K_{\mathrm{EJR2},u,\alpha}$)	demo_k_ejr2.py
exponentiated Jensen-Tsallis kernel-1 ($K_{\mathrm{EJT1},u,\alpha}$)	demo_k_ejt1.py
exponentiated Jensen-Tsallis kernel-2 ($K_{\mathrm{EJT2},u,\alpha}$)	demo_k_ejt2.py
conditional Shannon entropy [$H(\cdot\vert\cdot)$]	demo_h_shannon_cond.py
conditional Shannon mutual information [$I(\cdot\vert\cdot)$]	demo_i_shannon_cond.py
independence of y^m-s $\overset{?}{\Rightarrow} A\left(y^1,\ldots,y^M\right)=0$	demo_a_independence.py
independence of $\mathbf{y}^m$-s $\overset{?}{\Rightarrow} I\left(\mathbf{y}^1,\ldots,\mathbf{y}^M\right)=0$	demo_i_independence.py
$f_1=f_2 \overset{?}{\Rightarrow} D\left(f_1,f_2\right)=0$	demo_d_equality.py
approximation quality of incomplete Cholesky decomposition	demo_incomplete_cholesky.py
$f_1,\ldots,f_M \overset{?}{\Rightarrow} G=[G_{ij}]=[K(f_i,f_j)]_{i,j=1}^M$: positive semi-definite	demo_k_positive_semidefinite.py

Table 9: Top-middle: demos for analytical formula vs. estimated value (analytical_values subfolder); unconditional quantities (top), conditional ones (middle). 1st column: estimated quantity. 2nd column: .py. Bottom: independence demos, equality test, quality demo of incomplete Cholesky decomposition, positive semi-definiteness test for kernels on distributions (other subfolder). 1st column: task. 2nd column: .py.

A For Developers

Section A.1 is about the directory structure of the package. Section A.2 focuses on doctests. Adding new estimators is the topic of Section A.3. Passing parameters in certain meta-estimators is detailed in Section A.4.

A.1 Directory Structure

The `ite` package is organized as follows:

- `demos`: demos of the estimators.
  - `analytical_values`: analytical expressions (for a few distributions) vs. estimated values.
  - `other`: incomplete Cholesky decomposition, positive semi-definiteness of the Gram matrix defined by a kernel on distributions.
- `doc`: link to this manual.
- `ite`: contains the estimators themselves.
  - Directory `cost`:
    - In case of the
      1. unconditional quantities: the base estimators can be found in base_a.py (association measures), base_c.py (cross quantities), base_d.py (divergence measures), base_h.py (entropy), base_i.py (mutual information), base_k.py (kernels on distributions). The meta ones are in meta_a.py, meta_c.py, meta_d.py, meta_h.py, meta_i.py, meta_k.py.
      2. conditional ones: the meta estimators are in meta_h_cond.py (entropy), meta_i_cond.py (mutual information).
    - x_initialization.py, x_verification.py: classes to code up new estimators rapidly (initialization and verification routines).
    - x_factory.py: general module to invoke estimators.
    - x_analytical_values.py: analytical values for a various information theoretical quantities.
    - x_kernel.py: Kernel class.
    - x_python_to_matlab.py: Python ITE ↔ Matlab ITE correspondence (see Section B).
    - __init__.py: it makes the estimators available upon 'import ite'.
  - __init__.py: it loads the cost module upon 'import ite'.
  - shared.py: code shared by the estimators.

A.2 Running Doctests

Assumption: you have Nose installed.[6] Change to the main `ite` folder (containing the .txt-s, `doc`, `ite`, . . .), and issue the command

```
> nosetests --with-doctest -w ite    # run only doctests of ite/ite; 'nose' provides 'nosetests'
> nosetests --with-doctest base_a.py # after cd-ing to ite/ite/cost, run the doctests of a single
                                     # file (base_a.py)
> nosetests --with-doctest base_a.py:BASpearman1 # doctest of a specific class/function in base_a.py
> nosetests --with-doctest base_a                # the .py extension can be discarded
> nosetests --with-doctest base_a:BASpearman1    # -||-
```

A.3 Adding New Estimators

Upon creating a new estimator (H/I/D/A/C/K/condH/condI):

1. Recall the TypeXName naming convention (see Section 3.3).
2. The classes in Table 10 and Table 11 can be used for initialization and verification of the estimators.

   Notes:

   - 'InitX' is the default base class providing printing functionality (see below), and sets mult. Currently, multiplicative constants are considered to be relevant; in other words the default value of 'mult' is 'True'.[14]

     ```
     >>> import ite                      # load the ite package
     >>> co = ite.cost.BHShannon_KnnK() # initialize an entropy estimator
     >>> print(co)                      # print it(s parameters)
     ```

     This printing capability is what we used in Example 1.
   - 'VerCompSubspaceDims' is used for A/I estimators: an exception is raised if the subspace dimensions ($\{d_m\}_{m=1}^M$) and the dimension of the samples ($dim(\mathbf{y}_t)$) are not compatible.
   - 'VerEqualDSubspaces' guarantees in C/D/K estimators that an exception occurs if the dimensions of the samples from $\mathbf{y}^1$ and $\mathbf{y}^2$ are different.
   - 'InitBagGram' is the base class for *kernels on distributions* (empirically on bags of points), giving Gram matrix computation capability.
   - By 'InitKernel' one can build a kernel-based information theoretical estimator; it does not have to be a kernel on distributions. The currently implemented kernels are

     $$k_G(a,b) = e^{-\frac{\|a-b\|_2^2}{2\theta^2}}, \qquad k_e(a,b) = e^{-\frac{\|a-b\|_2}{2\theta^2}}, \qquad k_C(a,b) = \frac{1}{1+\frac{\|a-b\|_2^2}{\theta^2}},$$

     $$k_t(a,b) = \frac{1}{1+\|a-b\|_2^\theta}, \qquad k_p(a,b) = \left(\langle a,b\rangle + \theta\right)^p, \qquad k_r(a,b) = 1 - \frac{\|a-b\|_2^2}{\|a-b\|_2^2+\theta},$$

     $$k_i(a,b) = \frac{1}{\sqrt{\|a-b\|_2^2+\theta^2}}, \quad k_{M,\frac{3}{2}}(a,b) = \left(1+\frac{\sqrt{3}\|a-b\|_2}{\theta}\right)e^{-\frac{\sqrt{3}\|a-b\|_2}{\theta}},$$

     $$k_{M,\frac{5}{2}}(a,b) = \left(1+\frac{\sqrt{5}\|a-b\|_2}{\theta}+\frac{5\|a-b\|_2^2}{3\theta^2}\right)e^{-\frac{\sqrt{5}\|a-b\|_2}{\theta}}.$$
   - Notice the two *different* meanings of 'kernel' ('InitBagGram' vs 'InitKernel') in Table 10: The two meanings/capabilites can be used *independently*. For example, MKJS_DJS uses 'InitBagGram' but not 'InitKernel'; BKExpected relies on 'InitBagGram' and 'InitKernel'; BDMMD_UStat comes from 'InitKernel' but not from 'InitBagGram'.
3. A simplified calling syntax is provided for an A/I estimator with '$\forall d_m = 1$' constraint (see Section 3.3).
4. Forcing 'mult=True' in the children [see e.g., (60), (62), (74)] and passing parameters to them (such as in Table 12) are implemented where it is necessary.
5. If the name of the estimator is added to ite/ite/cost/__init__.py, it is loaded automatically upon 'import ite'.

A.4 Parameter Passing for (Certain) Meta Estimators

Certain meta estimators set others' parameters during the estimation; see Table 12 for a summary.

[14] Occasionally significant computation can be saved if these multiplicative factors do not matter.

Class(Parent)	Feature
InitX(object)	string representation, initialization: mult
InitKnnK(InitX)	string representation, initialization: mult, kNN ($S = \{k\}$)
InitKnnKiTi(InitX)	string representation, initialization: mult, k-NN ($S = \{k_i(T_i)\}$)
InitAlpha(InitX)	string representation, initialization: mult, $\alpha \neq 1$
InitUAlpha(InitAlpha)	string representation, initialization: mult, $\alpha \neq 1$, $u > 0$
InitKnnKAlpha(InitAlpha)	string representation, initialization: mult, kNN ($S = \{k\}$), $\alpha \neq 1$
InitKnnKAlphaBeta(InitKnnKAlpha)	string representation, initialization: mult, kNN ($S = \{k\}$), $\alpha \neq 1, \beta \neq 1$
InitKnnSAlpha(InitAlpha)	string representation, initialization: mult, generalized kNN ($S \subseteq \{1, \ldots, k\}$), $\alpha \neq 1$
InitKernel(InitX)	string representation, initialization: mult, kernel
InitEtaKernel(InitKernel)	string representation, initialization: mult, kernel, $\eta > 0$ (incomplete Cholesky decomposition)
InitBagGram(object)	Gram matrix computation for kernels on distributions (empirically on bags of points)

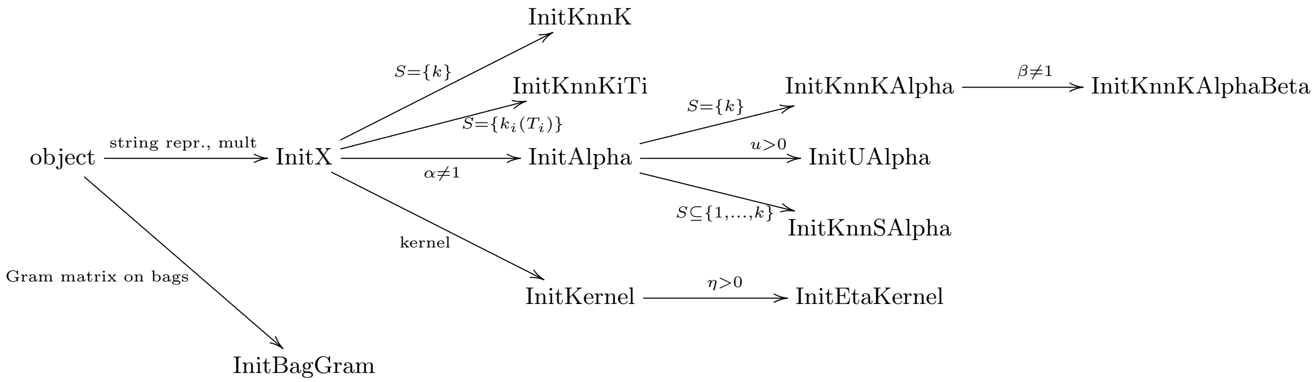

Table 10: Classes for initialization of the estimators (see x_initialization.py). 1st column: name of the class and its parent. 2nd column: its feature. The dependence of the classes and the additional features are schematically summarized in the diagram above.

Class	Feature
VerOneDSignal	$\mathbf{Y}_{1:T} \overset{?}{\in} \mathbb{R}^{T\times 1}$
VerOneDSubspaces	$d_1 = d_2 = \ldots = d_M = 1?$
VerCompSubspaceDims	$[d_1; \ldots; d_M], \mathbf{Y}_{1:T} \in \mathbb{R}^{T\times d} \overset{?}{\Rightarrow} d = \sum_{m=1}^M d_m$
VerSubspaceNumberIsK	$[d_1; \ldots; d_M] \overset{?}{\in} \mathbb{R}^K$
VerEqualDSubspaces	$\mathbf{Y}^1_{1:T_1} \in \mathbb{R}^{T_1\times d_1}, \mathbf{Y}^2_{1:T_2} \in \mathbb{R}^{T_2\times d_2} \overset{?}{\Rightarrow} d_1 = d_2$
VerEqualSampleNumbers	$\mathbf{Y}^1_{1:T_1} \in \mathbb{R}^{T_1\times d_1}, \mathbf{Y}^2_{1:T_2} \in \mathbb{R}^{T_2\times d_2} \overset{?}{\Rightarrow} T_1 = T_2$
VerEvenSampleNumbers	$\mathbf{Y}^1_{1:T} \in \mathbb{R}^{T\times d_1}, \mathbf{Y}^2_{1:T} \in \mathbb{R}^{T\times d_2} \overset{?}{\Rightarrow} 2\vert T$

Table 11: Classes for verification of the estimators (see x_verification.py). 1st column: name of the class. 2nd column: its feature.

Inherited parameter	Eq.	Cost name
$H_{\mathrm{T},\alpha} \xrightarrow{\alpha} H_{\mathrm{R},\alpha}$	(3)	MHTsallis_HR
$I_{\mathrm{R},\alpha} \xrightarrow{\alpha} D_{\mathrm{R},\alpha}$	(8)	MIRenyi_DR
$I_{\mathrm{R},\alpha} \xrightarrow{\alpha} H_{\mathrm{R},\alpha}$	(73)	MIRenyi_HR
$I_{\mathrm{T},\alpha} \xrightarrow{\alpha} D_{\mathrm{T},\alpha}$	(9)	MITsallis_DT
$D_{\mathrm{SB},\alpha} \xrightarrow{\alpha} D_{\mathrm{NB},\alpha}$	(36)	MDSymBregman_DB
$D^{\pi}_{\mathrm{JR},\alpha} \xrightarrow{\alpha} H_{\mathrm{R},\alpha}$	(42)	MDJR_HR
$D_{\mathrm{JT},\alpha} \xrightarrow{\alpha} H_{\mathrm{T},\alpha}$	(43)	MDJT_HT
$K_{\mathrm{JS}} \xrightarrow{\pi=\left[\frac{1}{2};\frac{1}{2}\right]} D^{\pi}_{\mathrm{JS}}$	(60)	MKJS_DJS
$K_{\mathrm{JT},\alpha} \xrightarrow{\alpha} H_{\mathrm{T},\alpha}$	(61)	MKJT_HT
$K_{\mathrm{EJS},u} \xrightarrow{\pi=\left[\frac{1}{2};\frac{1}{2}\right]} D^{\pi}_{\mathrm{JS}}$	(62)	MKExpJS_DJS
$K_{\mathrm{EJR1},u,\alpha} \xrightarrow{\alpha} H_{\mathrm{R},\alpha}$	(63)	MKExpJR1_HR
$K_{\mathrm{EJR2},u,\alpha} \xrightarrow{\pi=\left[\frac{1}{2};\frac{1}{2}\right],\alpha} D^{\pi}_{\mathrm{JR},\alpha}$	(64)	MKExpJR2_DJR
$K_{\mathrm{EJT1},u,\alpha} \xrightarrow{\alpha} H_{\mathrm{T},\alpha}$	(65)	MKExpJT1_HT
$K_{\mathrm{EJT2},u,\alpha} \xrightarrow{\alpha} D_{\mathrm{JT},\alpha}$	(66)	MKExpJT2_DJT

Table 12: Parameter passing in meta estimators. Notation $X \xrightarrow{z} Y$: the X meta method sets the z parameter(s) of the Y estimator. 2nd column: the equation describing this action. 3rd column: cost name.

B Python ITE ↔ Matlab ITE

Python cost names with Matlab equivalents (when it exists) are summarized in ite/ite/cost/x_python_to_matlab.py:

- Python → Matlab cost name transition: see dictionary `dict_X_PythonToMatlab`, where $X \in \{A, C, D, H, I, K, condH, condI\}$.
- Matlab → Python conversion, given a cost type X: see `dict_X_MatlabToPython`.

The equations of the estimators can be looked up by this correspondence and the Matlab ITE documentation.[15]

C For Mathy People: Axioms of Concordance and Dependence

This section summarizes the axiomatic formulations of concordance (Def. 1, 2, 3) and dependence (Def. 4, 5).

Definition 1 (concordance ordering) *In two dimensions ($d = 2$) a copula C_1 is said to be smaller than the copula C_2 ($C_1 \prec C_2$) [4], if*

$$C_1(\mathbf{u}) \leq C_2(\mathbf{u}), \quad \left(\forall \mathbf{u} \in [0,1]^2\right). \tag{77}$$

This pointwise partial ordering on the set of copulas is called concordance ordering. *In the general ($d \geq 2$) case, a copula C_1 is said to be smaller than the copula C_2 ($C_1 \prec C_2$) [2], if*

$$C_1(\mathbf{u}) \leq C_2(\mathbf{u}) \text{ and } \bar{C}_1(\mathbf{u}) \leq \bar{C}_2(\mathbf{u}) \quad \left(\forall \mathbf{u} \in [0,1]^d\right). \tag{78}$$

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.*
- *The two definitions [(77), (78)] coincide only in the two-dimensional ($d = 2$) case.*

Definition 2 (measure of concordance [5, 3, 4]) *A numeric measure of association κ 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:*

[15] Available at https://bitbucket.org/szzoli/ite/downloads; see Section E (Estimation Formulas).

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

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

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

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

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

A6. Coherence*: if* $C_1 \prec C_2$*, then* $\kappa(C_1) \leq \kappa(C_2)$*.*[16]

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

Note: the properties in brackets can be derived from the others.

Definition 3 (multivariate measure of concordance [1, 7]) *A* multivariate measure of concordance *is a function* κ *that assigns to every continuous random variable* $\mathbf{y}$ *a real number and satisfies the following requirements:*

B1. Normalization*:*

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

B1b : $\kappa(y^1, \dots, y^d) = 0$ *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* $\lim_{t\to\infty} \kappa(\mathbf{y}_t) = \kappa(\mathbf{y})$*. [In terms of copulas:* $\lim_{t\to\infty} C_t = C$ *(uniformly)* $\Rightarrow \lim_{t\to\infty} \kappa(C_t) = \kappa(C)$*.]*

B4. Permutation invariance*: if* $\{i_1, ..., i_d\}$ *is permutation of* $\{1, \dots, d\}$*, then* $\kappa(y^{i_1}, \dots, y^{i_d}) = \kappa(y^1, \dots, y^d)$*.*

B5. Duality*:* $\kappa(-y^1, \dots, -y^d) = \kappa(y^1, \dots, y^d)$*.*

B6. Reflection symmetry property*:* $\sum_{\epsilon_1, \dots, \epsilon_d = \pm 1} \kappa(\epsilon_1 y^1, \dots, \epsilon_d y^d) = 0$*, 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* $\mathbf{y}$ $r_{d-1}\kappa(y^2, \dots, y^d) = \kappa(y^1, \dots, y^d) + \kappa(-y^1, \dots, y^d)$*.*

Definition 4 (measure of dependence) *[4] defines 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* (y^1, y^2) *pair.*

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

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

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

C5. Strictly monotone functional dependence*:* $\kappa(y^1, y^2) = 1$ *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* $\kappa(y^1, y^2) = \kappa(f_1(y^1), f_2(y^2))$*.*

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

Definition 5 (multivariate measure of dependence) *A real-valued function* κ *is called a* measure of dependence *[8] for a d-dimensional random variable if it satisfies the following properties:*

[16] Hence the name concordance ordering.

D1. ***Domain:*** *κ is defined for every continuous random variable $\mathbf{y}$.*

D2. ***Permutation invariance:*** *if $\{i_1, ..., i_d\}$ is permutation of $\{1, \dots, d\}$, then $\kappa\left(y^{i_1}, \dots, y^{i_d}\right) = \kappa\left(y^1, \dots, y^d\right)$.*

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

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

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

D6. ***Invariance to strictly monotone functions:*** *If $f_1, \dots, f_d$ are all strictly increasing functions, then $\kappa\left(y^1, \dots, y^d\right) = \kappa\left(f_1\left(y^1\right), \dots, f_d\left(y^d\right)\right)$.*

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

References

[1] Ali Dolati and Manuel Úbeda-Flores. On measures of multivariate concordance. *Journal of Probability and Statistical Science*, 4:147–164, 2006.

[2] Harry Joe. Multivariate concordance. *Journal of Multivariate Analysis*, 35:12–30, 1990.

[3] Roger B. Nelsen. *Distributions with Given Marginals and Statistical Modelling*, chapter Concordance and copulas: A survey, pages 169–178. Kluwer Academic Publishers, Dordrecht, 2002.

[4] Roger B. Nelsen. *An Introduction to Copulas*. Springer, 2006.

[5] Marco Scarsini. On measures of concordance. *Stochastica*, 8:201–218, 1984.

[6] Zoltán Szabó. Information theoretical estimators toolbox. *Journal of Machine Learning Research*, 15:283–287, 2014.

[7] M. D. Taylor. Multivariate measures of concordance. *Annals of the Institute of Statistical Mathematics*, 59:789–806, 2007.

[8] Edward F. Wolff. N-dimensional measures of dependence. *Stochastica*, 4:175–188, 1980.