INFOTOPO Documentation

Release stable

Oct 11, 2020

User Guide / Tutorial:

InfoTopo: Topological Information Data Analysis. Deep statistical unsupervised and supervised learning.

InfoTopo is a Machine Learning method based on Information Cohomology, a cohomology of statistical systems [0,1,8,9]. It allows to estimate higher order statistical structures, dependences and (refined) independences or generalised (possibly non-linear) correlations and to uncover their structure as simplicial complex. It provides estimations of the basic information functions, entropy, joint and condtional, multivariate Mutual-Informations (MI) and conditional MI, Total Correlations… The package was written to be fully compliant with scikit learn tools, objects and nomenclature. InfoTopo is at the cross-road of Topological Data Analysis, Deep Neural Network learning, statistical physics and complex systems:

1. With respect to Topological Data Analysis (TDA), it provides intrinsically probabilistic methods that does not assume metric (Random Variable's alphabets are not necessarilly ordinal) [2,3,6]. It also provide a quantification of higher order statistical interactions that cannot be detected by pairwise relations or methods based on Vietoris-Rips complexes.
2. With respect to Deep Neural Networks (DNN), it provides a simplical complex constrained DNN structure with topologically derived unsupervised and supervised learning rules (forward propagation, differential statistical operators). The neurons are random Variables, the depth of the layers corresponds to the dimensions of the complex [3,4,5].
3. With respect to statistical physics, it provides generalized correlation functions, free and internal energy functions, estimations of the n-body interactions contributions to energy functional, that holds in non-homogeous and finite-discrete case, without mean-field assumptions. Cohomological Complex implements the minimum free-energy principle. Information Topology is rooted in cognitive sciences and computational neurosciences, and generalizes-unifies some consciousness theories [5].
4. With respect to complex systems studies, it generalizes complex networks and Probabilistic graphical models to higher degree-dimension interactions [2,3].

(5.) To add just some other buzz words, please be sure that the methods presented here could fully pertain to "explainable AI", although just like mathematic it has nothing artificial, as long as mathematic will be the language of nature and it does not guarantees any inteligence nor its converse, this is indeed up to the use you will make of it.

It assumes basically:

1. a classical probability space (here a discrete finite sample space), geometrically formalized as a probability simplex with basic conditionning and Bayes rule and implementing
2. a complex (here simplicial) of random variable with a joint operators
3. a quite generic coboundary operator (Hochschild, Homological algebra with a (left) action of conditional expectation)

The details for the underlying mathematics and methods can be found in the papers:

[0] Manin, Y., Marcolli, M., Homotopy Theoretic and Categorical Models of Neural Information Networks, 2020, arXiv:2006.15136, PDF-0

[1] Vigneaux J., Topology of Statistical Systems. A Cohomological Approach to Information Theory. Ph.D. Thesis, Paris 7 Diderot University, Paris, France, June 2019. PDF-1

[2] Baudot P., Tapia M., Bennequin, D. , Goaillard J.M., Topological Information Data Analysis. 2019, Entropy, 21(9), 869 PDF-2

[3] Baudot P., The Poincaré-Shannon Machine: Statistical Physics and Machine Learning aspects of Information Cohomology. 2019, Entropy , 21(9), PDF-3

[4] Baudot P. , Bernardi M., The Poincaré-Boltzmann Machine: passing the information between disciplines, ENAC Toulouse France. 2019 PDF-4

[5] Baudot P. , Bernardi M., Information Cohomology methods for learning the statistical structures of data. DS3 Data Science, Ecole Polytechnique 2019 PDF-5

[6] Tapia M., Baudot P., Dufour M., Formizano-Treziny C., Temporal S., Lasserre M., Kobayashi K., Goaillard J.M.. Neurotransmitter identity and electrophysiological phenotype are genetically coupled in midbrain dopaminergic neurons. Scientific Reports. 2018. PDF-6

[7] Baudot P., Elements of qualitative cognition: an Information Topology Perspective. Physics of Life Reviews. 2019. extended version on Arxiv. PDF-7

[8] Baudot P., Bennequin D., The homological nature of entropy. Entropy, 2015, 17, 1-66; doi:10.3390. PDF-8

[9] Baudot P., Bennequin D., Topological forms of information. AIP conf. Proc., 2015. 1641, 213. PDF-9

You can find the software on github.

The previous version of the software INFOTOPO : the 2013-2017 scripts are available at Github infotopo

Installation

PyPI install, presuming you have numpy and networkx installed:

```
pip install infotopo
```

CHAPTER 1

How to Use InfoTopo

Infotopo is a general Machine Learning set of tools gathering Topology (Cohomology and Homotopy), statistics and information theory (information quantifies statistical structures generically) and statistical physics. It provides a matheamticaly formalised expression of deep network and learning, and propose anuspervised or supervised learning mode (as a special case of the first). It allows a simple and autonatic exploration of the data structures, dimension reduction and supervised or unsupervised classification.

The raw methods are computationnally consuming due to the intrinsic combinatorial nature of the topological tools, even in the simplest case of a simplicial case (the general case is based on the much broader partition combinatorics) the computational complexity is of the order of 2^n . As a consequence, an important part of the tools and methods are dedicated to overcome this extensive computation. Among the possible strategies and heuristics used or currently developped, are:

_ restrict to simplicial cohomology and combinatorics (done here).

_ possible exploration of only the low dimensional structures (done here).

_ possible exploration of only most or least informative paths (done here).

_ possible restriction 2nd degree-dimension statistical interactions: what is computed here is the equivalent of the Cech complex (with all degree- dimension computed), and such restriction is equivalent to computing the Vietoris-Rips complex (in development).

_ compute on GPU and Tensorflow version (in development).

As a result, for this 0.1 version of the software, and for computation with commercial average PC, we recommand to analyse up to 20 variables (or dimensions) at a time in the raw brut-force approach (see performance section).

We now present some basic example of use, inspiring our presentation from the remarkable presentation of UMAP by McInnes. We first import some few tools: some of the datasets available in sklearn, seaborn to visualise the results, and pandas to handle the data.

```
from sklearn.datasets import load_iris, load_digits, load_boston, load_diabetes
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import timeit
```

```
sns.set(style='white', context='notebook', rc={'figure.figsize':(14,10)})
```

1.1 Iris data

1.1.1 Iris dataset

The first example of dataset application we will present is the iris dataset. It is a very small dataset composed of 4 Random-Variables or dimensions that quantify various petals and sepals observables of 3 different species of Iris flowers, like petal length, for 150 flowers or points (50 for each species). In the context of Infotopo it means that dimension_tot = 4 and sample_size = 150 (we consider all the points), and as the dimension of the data set is small we will make the complete analysis of the simplicial structure of dependencies by setting the maximum dimension of analysis to dimension_max = dimension_tot. We also set the other parameters of infotopo to approriate, as further explained. We can load the iris dataset from sklearn.

```
iris = load_iris()
iris_df = pd.DataFrame(iris.data, columns = iris.feature_names)
print(iris.DESCR)

dimension_max = iris.data.shape[1]
dimension_tot = iris.data.shape[1]
sample_size = iris.data.shape[0]
nb_of_values =9
forward_computation_mode = False
work_on_transpose = False
supervised_mode = False
sampling_mode = 1
deformed_probability_mode = False
```

```
Iris Plants Database
====================

Notes
-----
Data Set Characteristics:
    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica
    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
```

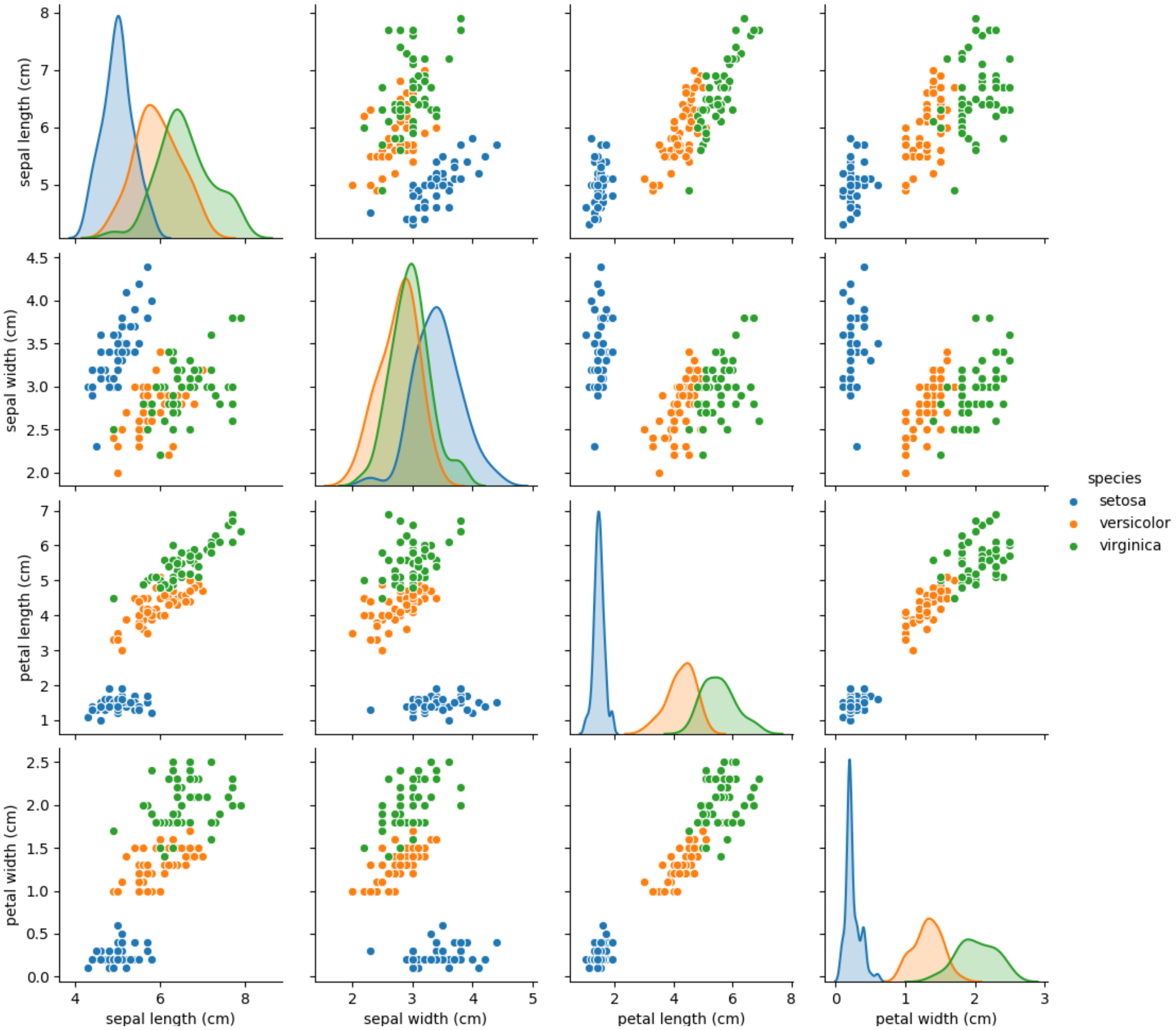
(continues on next page)

(continued from previous page)

```
    petal width:    0.1  2.5   1.20  0.76     0.9565  (high!)
    ============== ==== ==== ======= ===== ====================

    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988

This is a copy of UCI ML iris datasets.
http://archive.ics.uci.edu/ml/datasets/Iris

The famous Iris database, first used by Sir R.A Fisher

This is perhaps the best known database to be found in the
pattern recognition literature.  Fisher's paper is a classic in the field and
is referenced frequently to this day.  (See Duda & Hart, for example.)  The
data set contains 3 classes of 50 instances each, where each class refers to a
type of iris plant.  One class is linearly separable from the other 2; the
latter are NOT linearly separable from each other.

References
----------
   - Fisher,R.A. "The use of multiple measurements in taxonomic problems"
     Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to
     Mathematical Statistics" (John Wiley, NY, 1950).
   - Duda,R.O., & Hart,P.E. (1973) Pattern Classification and Scene Analysis.
     (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.
   - Dasarathy, B.V. (1980) "Nosing Around the Neighborhood: A New System
     Structure and Classification Rule for Recognition in Partially Exposed
     Environments".  IEEE Transactions on Pattern Analysis and Machine
     Intelligence, Vol. PAMI-2, No. 1, 67-71.
   - Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE Transactions
     on Information Theory, May 1972, 431-433.
   - See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al"s AUTOCLASS II
     conceptual clustering system finds 3 classes in the data.
   - Many, many more ...
```

As visualizing data in 4 dimensions or more is hard or not possible, we can first plot all the pairwise scatterplot matrix to present the pairwise correlations and dependencies between the variables, using Seaborn and pandas dataframe.

```
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
iris_df['species'] = pd.Series(iris.target).map(dict(zip(range(3),iris.target_names)))
sns.pairplot(iris_df, hue='species')
plt.show()
```



All those 2D views gives a rought but misleading idea of what the data looks like in high dimensions since, as we will see, some fully emergent statistical dependences (called synergy in the original work of Bialek's team) can appear in higher dimension which are totally unobservable in those 2D views. However such 2D views gives a fair visual estimation of how much each pairs of variale covary, the correlation coefficient and its generalization to non-linear relations, the pairwise Mutual Information (I2). In Topological Data Analysis (TDA) terms, it gives rought idea of what the skeleton of a Vietoris-Rips (information or correlation) complex of the data could be. We will see how to go beyond this pairwise statistical interaction case, and how we can unravel some purely emergent higher dimensional interations. Along this way, we will see how to compute and estimate all classical information functions, multivariate Entropies, Mutual Informations and Conditional Entropies and Mutual Informations.

1.1.2 Entropy

To use infotopo we need to first construct a infotopo object from the infotopo package. This makes a lot of same word, information is a functor, a kind of general application or map, that could be either a function or a class. So let's first import the infotopo library, we a set of specifications of the parametters (cf. section parameters, some of them like dimension_max = dimension_tot and sample_size have been fixed previously to the size of the data input matrix).

```
import infotopo
```

```
information_topo = infotopo.infotopo(dimension_max = dimension_max,
                            dimension_tot = dimension_tot,
                            sample_size = sample_size,
                            work_on_transpose = work_on_transpose,
                            nb_of_values = nb_of_values,
                            sampling_mode = sampling_mode,
                            deformed_probability_mode = deformed_probability_mode,
                            supervised_mode = supervised_mode,
                            forward_computation_mode = forward_computation_mode)
```

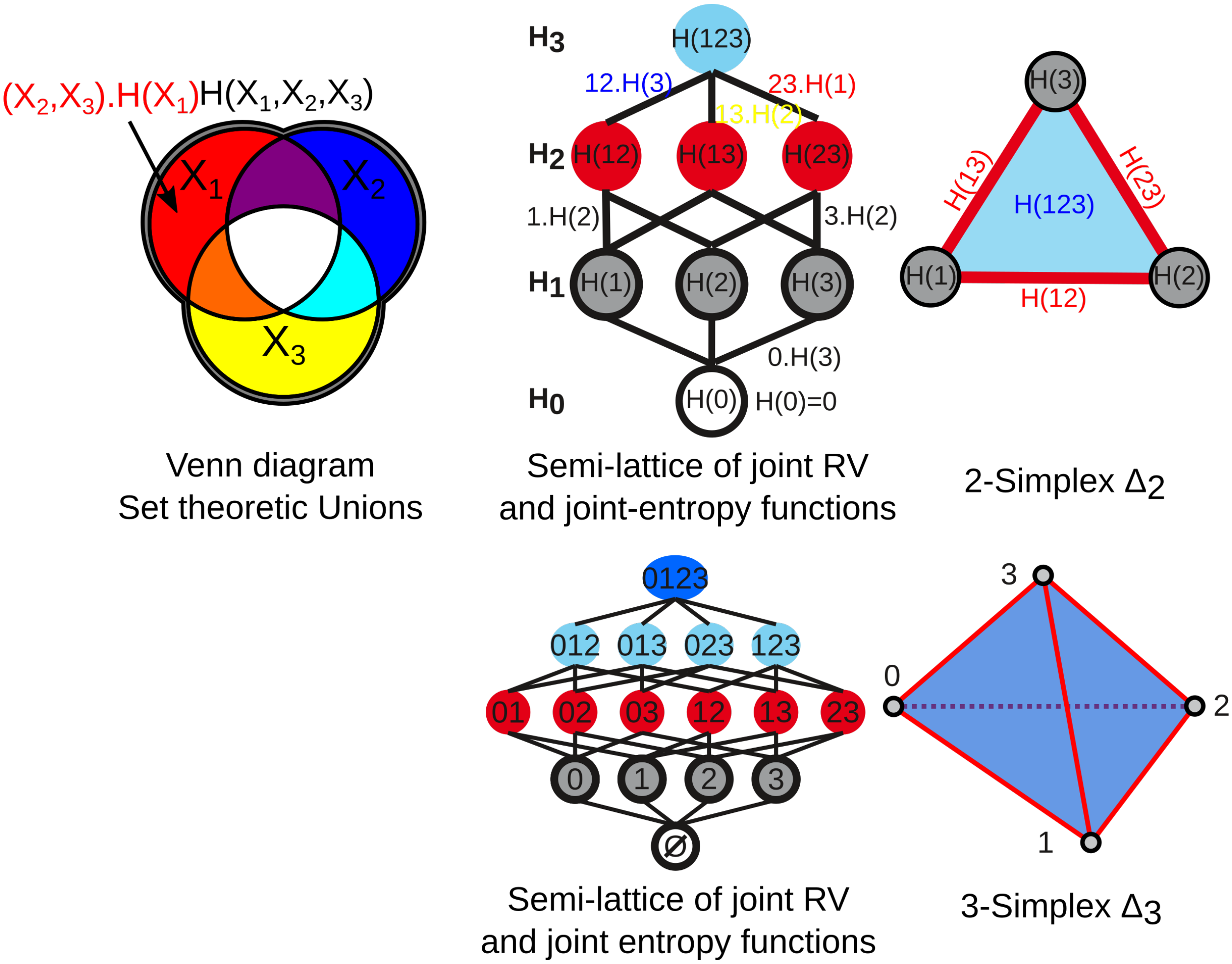
Now we will compute all the simplicial semi-lattice of marginal and joint-entropy, that contains 2^n elements including the unit 0 reference measure element. The marginal H_1 entopies are defined as classicaly by Shannon :

$$H_1 = H(X_j; P) = k \sum_{x \in [N_j]} p(x) \ln p(x)$$

and the multivariate joint-entropies H_k just generalise the preceding to k variables:

$$H_k = H(X_1, ..., X_k; P) = k \sum_{x_1, ..., x_k \in [N_1 \times ... \times N_k]}^{N_1 \times ... \times N_k} p(x_1 .....x_k) \ln p(x_1 .....x_k)$$

The figure below give the usual Venn diagrams representation of set theoretic unions and the corresponding semi-lattice of joint Random Variables and Joint Entropies, together with its correponding simplicial representation, for 3 (top) and 4 variables-dimension (bottom, the case of the iris dataset with 2 power 4 joint random variables). This correspondence of joint-information with the semi-lattice of union was formalized by Hu Kuo Ting . The edges of the lattice are in one to one correspondence with conditional entropies.



Venn diagram
Set theoretic Unions

Semi-lattice of joint RV
and joint-entropy functions

2-Simplex Δ_2

Semi-lattice of joint RV
and joint entropy functions

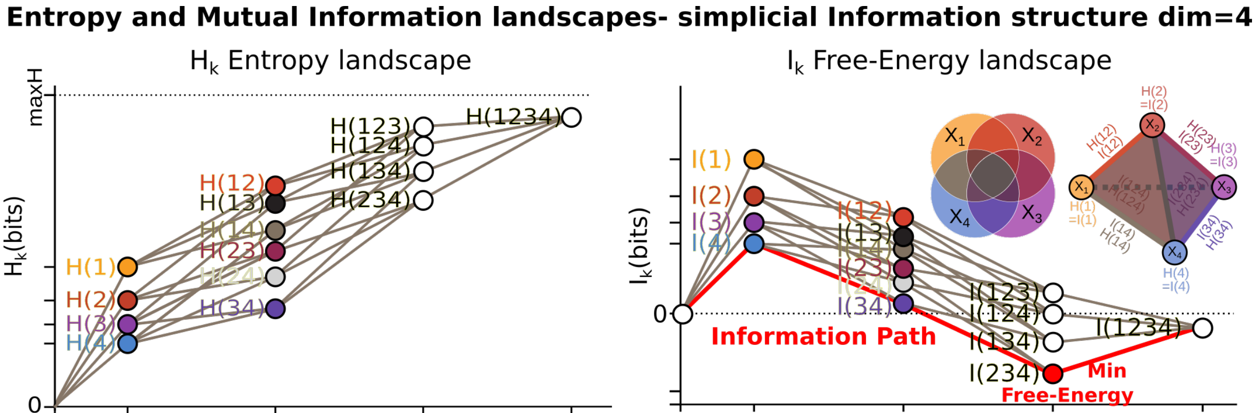
3-Simplex Δ_3

To do this we will call simplicial_entropies_decomposition, that gives in output all the joint entropies in the form of a dictionary with keys given by the tuple of the joint variables (ex: (1,3,4)) and with values the joint or marginal entropy in bit (presented below).

```
Nentropie = information_topo.simplicial_entropies_decomposition(iris.data)
```

```
{(4,): 2.9528016441309237, (3,): 2.4902608474907497, (2,): 2.5591245822618114, (1,):␣
↪2.8298425472847066, (3, 4): 3.983309507504916, (2, 4): 4.798319817958397, (1, 4): 4.
↪83234271597051, (2, 3): 4.437604597473526, (1, 3): 4.2246575340121835, (1, 2): 4.
↪921846615158947, (2, 3, 4): 5.561696151051504, (1, 3, 4): 5.426426190681815, (1, 2,␣
↪4): 6.063697650692486, (1, 2, 3): 5.672729631265195, (1, 2, 3, 4): 6.
↪372515544003377}
```

Such dictionary is hard to read; to allow a relevant visualization of the the simplicial entropy structure, the function simplicial_entropies_decomposition also plots the Entropy landscapes. Entropy landscapes provides a representation of the lattice of joint (H_k) and conditional entropies (noted as the action of Y $Y.H_k$, for $H(X_1, ..., X_k|Y)$) that ranks the joint variables as a function of their entropy value and of the rank-dimensions as illustrated in the figure below:



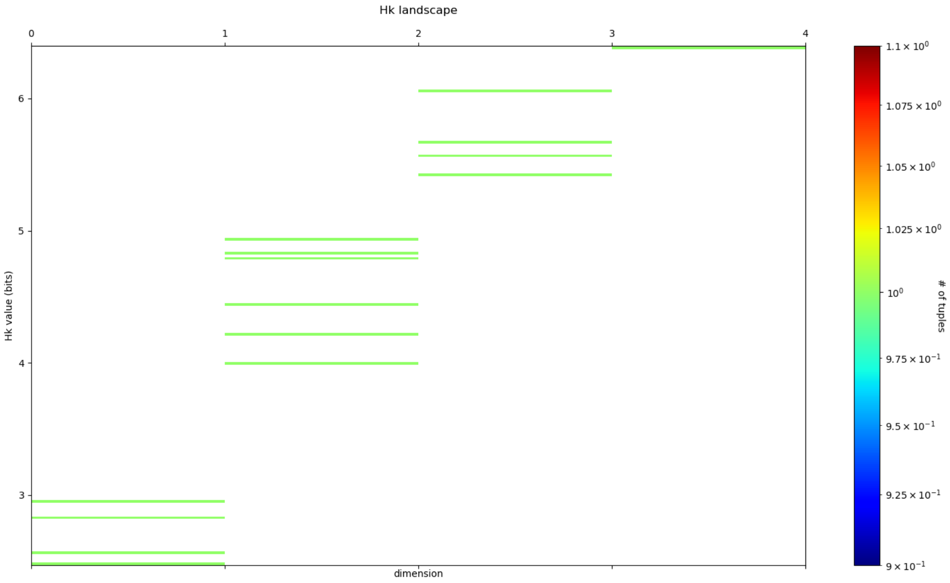
An Entropy of Information Path is a sequence of inclusive tuples of increasing dimensions and follows the edges of the semi-lattice, and the slope of such a path is exactly minus the conditional-entropy, as a basic representation of the fundamental chain rule of Entropy.

While the total dimension n (dimension_tot) of the analysis increases, the number of subsets of k variables (or k-tuples) increases combinatorially, following the binomial coefficient C(n,k). It hence becomes rapidly fully impractical to vizualize, plot and to differentiate the C(n,k) values of entropy obtained in dimension k. The Entropy landscapes hence plot the histograms of entropy values as a function of the dimension-rank k, and the number of bins of the histograms is imposed by the parameter nb_bins_histo. The count of the number of subsets having entropy values in the bin range of the histograms is represented by a color code in the entropy landscapes. Hence, Entropy Landscapes shall be understood as (unormalised..but it could be normalised) entropy measure densities histograms (there is interesting further theoretical and applied development upon this point, since entropy functions obey axioms of measure: one could legitamely investigate entropies of entropies, a kind of complexity of information landscapes, see Hsu et al.).

To plot the Entropy Landscapes and the distribution of entropy values for each dimension-rank k, we use the “entropy_simplicial_lanscape” command as following:

```
information_topo.entropy_simplicial_lanscape(Nentropie)
```

On the example of Iris dataset, the Entropy Landscape we obtain look like this:



In this low dimensional case (dimension_tot = 4), the landscapes are very low informative (poor information structure) and the histrograms have low meaning, since there is only one subset-k-tuple per bin value, and hence only one color (here the green value of 1). The Entropy Landscape themselfs are quite poor in information, joint-entropy is monotonically increasing along entropy path, a direct consequence of conditional-entropy positivity (concavity argument) which is moreover the basic fact at the origin of the basic topological expression of the 2nd law of thermodynamic [3]. As a consequence, we usually do not uncover a lot of usefull information on the datas structure from those Entropy Landscape, at the exception of curse of dimensionality quantification and in some cases, (assymptotic) entropy rates (to do). Basically, joint-entropy quantifies "randomness" (in a non formal definition of the word), uncertainty, or how much the data points spreads in the dimensions of the variables. Hence low entropies shall be intrepreted as "localised" densities of data points or sparsness of the probability density histograms (also not in the usual kurtosis sens).

In any entropy or information function estimation, it is necessary to check that the number of sample is sufficient to provide a faithfull estimate, to avoid the sampling problem also called "curse of dimension". The command "entropy_simplicial_lanscape" also computes the maximal dimension above which the estimation becomes too inacurate and shall not be interpreted. This is explained in more details in the section "curse_of_dimension_and_statistical_dependencies_test".

1.1.3 Mutual Information

Now, let's have a look at the statistical dependencies structures in the dataset by computing the Mutual-Information lanscapes which principle is depicted in the preceding figure and that basically plots k-dimensional multivariate Mutual Informations (I_k) in the same way as Entropy Landscapes. Pairwise Mutual Information I_2 is defined as usual following Shannon :

$$I_2 = I(X_1; X_2; P) = k \sum_{x_1,x_2 \in [N_1 \times N_2]}^{N_1 \times N_2} p(x_1.x_2) \ln \frac{p(x_1)p(x_2)}{p(x_1.x_2)}$$

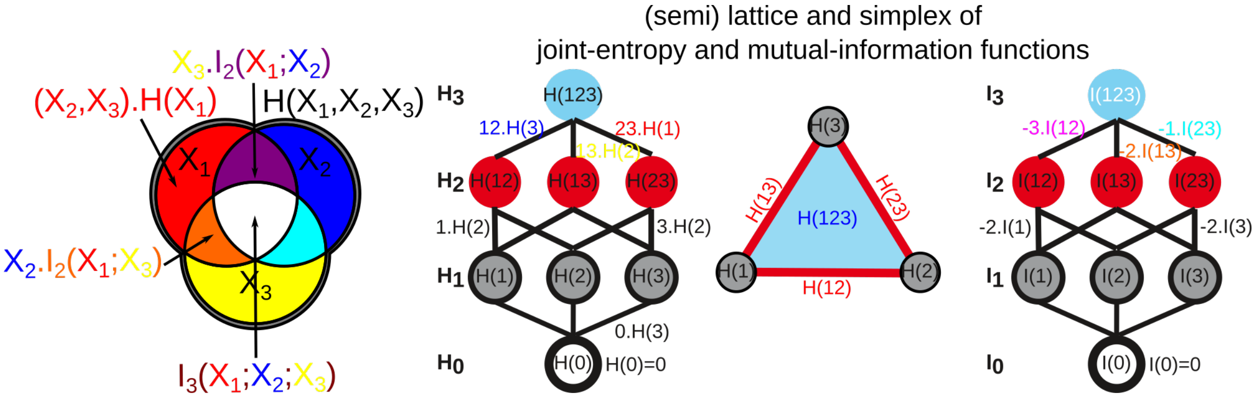
They generalize to the multivariate mutual informations, I_k, as alternated functions of entropies, following McGill and Hu Kuo Ting.

$$I_k(X_1, ..., X_k; P) = \sum_{i=1}^{k} (-1)^{i-1} \sum_{I \subset [k]; card(I)=i} H_i(X_I; P)$$

For example: .. math:

```
`I_3=H(X_1)+H(X_2)+H(X_3)-H(X_1,X_2)-H(X1,X_3)-H(X_2,X_3)+H(X_1,X_2,X_3)`:
```

Hu Kuo Ting showed the correspondence of I_k with set intersections semi-lattice (of finite measurable functions), and we hence have just like with entropy the following information structure, corresponding to intections on Venn diagrams:



The other functions that quantifies multivariate depence are Total Correlations, G_k (Watanabe , see section diabetes data) , or total free energy, or Integrated Information (Tononi and Edelman) which are the Kullback-Leibler Divergence between the full joint-entropy and its marginals product, for example, $G_3 = H(X_1) + H(X_2) + H(X_3) - H(X_1, X_2, X_3)$:

$$G_k = G_k(X_1; ...X_k; P) = \sum_{i=1}^{k} H(X_i) - H(X_1; ...X_k)$$

Whereas, G_k quantifies the total interactions, I_k quantify the contribution of the kth interaction. Notably, we have the theorems that state that n variables are independent if and only if $G_n = 0$, and n variables are independent if and only if all the $2^n - n - 1$ I_k functions with $k \geq 2$ are null (e.g. I_k provides a refined independence measure PDF). In contrast with G_k, I_k can be negative for $k \geq 3$, a phenomenon called synergy in the original study of Brenner et al. Considering the old goal of expressing all of physics in terms of information, following Brillouin, Jaynes, Wheeller (...), for $k\ geq\ 2$, G_k corresponds to a Free-Energy functional of a k interacting body system, while the I_k quantifies the contribution of the k-bodies interaction to this total free energy. The I_1 component is the internal energy:

$$H_k(X_1, .., X_k; P_N) = E(X_1, .., X_k; P_N) - G(X_1, .., X_k; P_N) = E - G$$

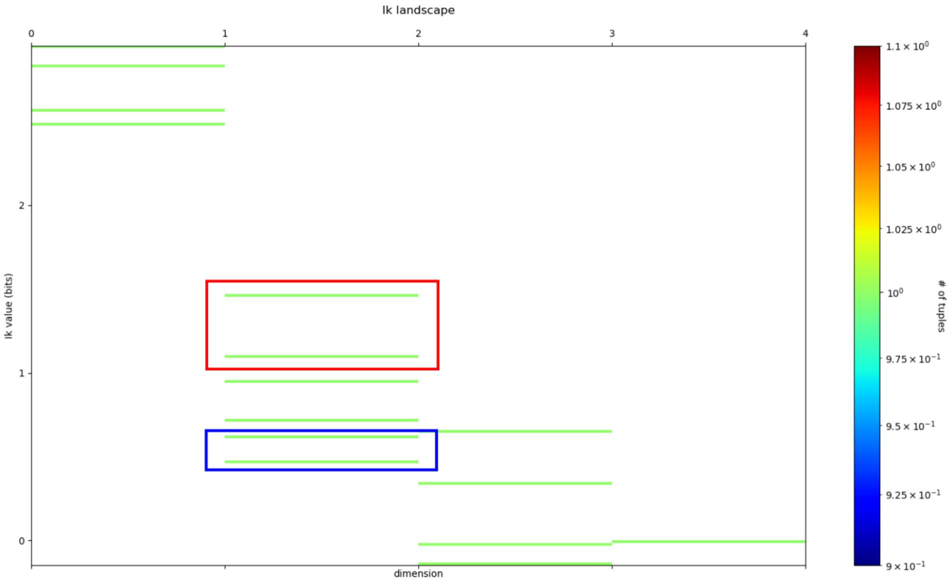
The Free-energy decomposes itself as an alternated sum of I_k :

$$G_k = \sum_{i=2}^{k} (-1)^i \sum_{I \subset [n]; card(I)=i} I_i(X_I; P)$$

To plot the Information Landscapes and the distribution of I_k values for each dimension-rank k, we use the "entropy_simplicial_lanscape" command as following:

```
information_topo.mutual_info_simplicial_lanscape(Ninfomut)
```

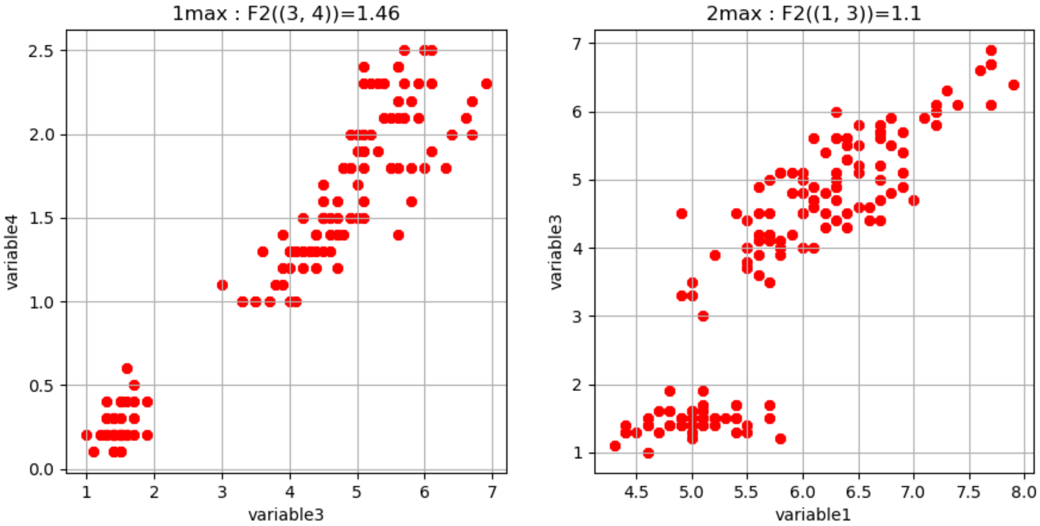
On the example of Iris dataset, the Entropy Landscape we obtain look like this:



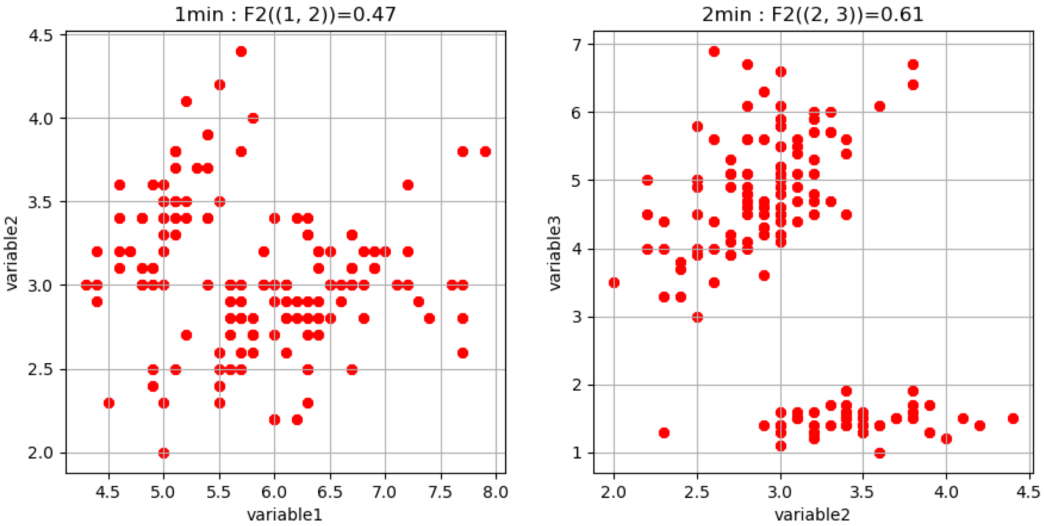
To obtain the first m k-tuples with maximum and minimum value in dimension k, and if the dimension is 2,3 or 4 plot the data points in the corresponding k-subspace (the 4th dimension is represented by a color code), we use the "display_higher_lower_information". For exmaple, plotting the 2 first maximum and minimum in dimension (framed in red and blue respectively in the last figure), we use the following command:

```
information_topo = infotopo(dim_to_rank = 2, number_of_max_val = 2)
dico_max, dico_min = information_topo.display_higher_lower_information(Ninfomut, dataset)
```

On the example of Iris dataset, we obtain the two pairs of variables (3,4) and (1,3) that are the most statistically dependent ("correlated"):



And we obtain the two pairs of variables (1,2) and (2,3) that are the less statistically dependent ("uncorrelated"):



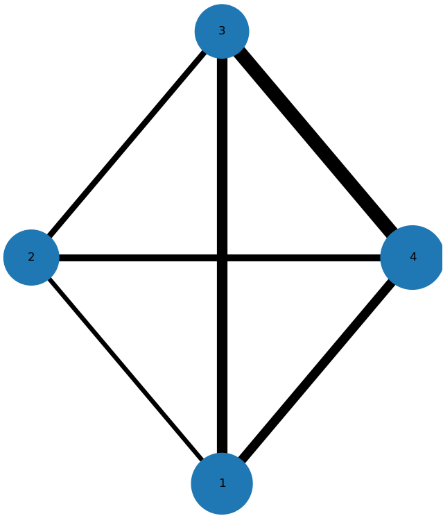
Whenever the dimension to study is more than 4, the function only retreives the dictionaries of the first maximum and minimum tuples (to print).

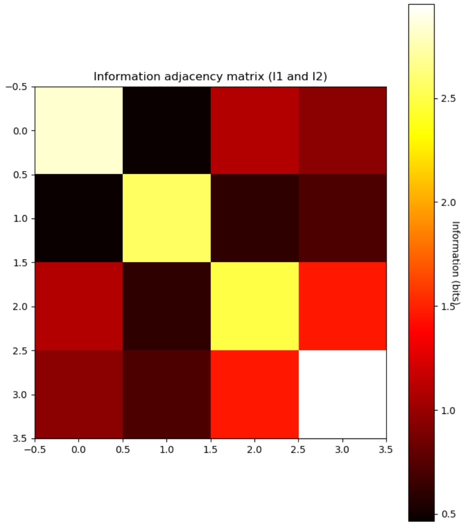
1.1.4 Information Networks

In biology (e.g "omic"), neuroscience (e.g "neural network") and social science (e.g "social network"), it is common and helpfull to conceive and visualize the one and two dimensional results as (first degree) networks. To visualize the Information Networks, we use the "mutual_info_pairwise_network" as following:

```
adjacency_matrix_mut_info = information_topo.mutual_info_pairwise_network(Ninfomut)
```

The area of each vertex is a function of the marginals information $H_1 = I_1$ and the thickness of the edges is a function of the pairwise mutual information or total correlation $I_2 = G_2$. On Iris dataset, it gives:





The adjacency matrix of information have the marginals informations $H_1 = I_1$ in its diagonal and is symmetric with respect to the diagonal as the result of the commutativity of the join-variables and mutual-variables operation in classical information theory (classical is by opposition with quantum information theory). Compared to usual distance matrix (with given metric) computed in machine learning (for clustering or classifications), the I_k are not metric (e.g. non zero diagonal and no triangle inequality), we will introduce to information metric in the next stepps. With such Matrix it is possible to apply some usual computational persistence homology tools like Mapper scikit-tda (created by Singh, Mémoli, and Carlsson) and to build what could be called an "informational Vietoris-Ripps complex". In the context of Morse theory, information landscapes consider infomation functions themselfs as height or "Morse" functions. However there is likely a much more fundamental application of persistence theory in the construction of a local probability density estimation (to be done). I_k with $k \geq 3$ can be repesented in an analgous way using k-cliques as acheived in Tapia & al 2018 (to be done in the package). They shall be represented using k-tensor formalism. In the context of complex networks studies those higher I_k with $k \geq 3$ correspond to hypergraphs or multiplex or multilayer networks The raw result obtained here is a fully connected network, but one can obtain a sparse matrix and a sparsely connected network by thresholding the I_k with a with fixed p-value, using the exact statistical dependence test implemented in the package.

We begin to see that Homology provides a wide generalisation of complex networks (a 1-complex, that is a graph) to higher interactions structures.

1.2 Diabetes data

1.2.1 Diabetes dataset

The Iris dataset and its associated information landsacpes are in too low dimension to appreciate all the interest of the methods in higher dimensions, so lets turn to larger dimensional classical machine learning dataset: Diabetes dataset. This dataset is kindly also furnished by scikitlearn, and we load it with the same methods as previously:

```
dataset = load_diabetes()
dataset_df = pd.DataFrame(dataset.data, columns = dataset.feature_names)
dimension_max = dataset.data.shape[1]
dimension_tot = dataset.data.shape[1]
sample_size = dataset.data.shape[0]
nb_of_values = 9
forward_computation_mode = False
work_on_transpose = False
supervised_mode = False
sampling_mode = 1
deformed_probability_mode = False
dataset_df = pd.DataFrame(dataset.data, columns=dataset.feature_names)
```

This dataset contains 10 variables-dimensions for a sample size (number of points) of 442 and a target (label) variable which quantifies diabetes progress. The ten variables are [age, sex, body mass index, average blood pressure, T-Cells, low-density lipoproteins, high-density lipoproteins, thyroid stimulating hormone, lamotrigine, blood sugar level] in this order.

1.2.2 Entropy

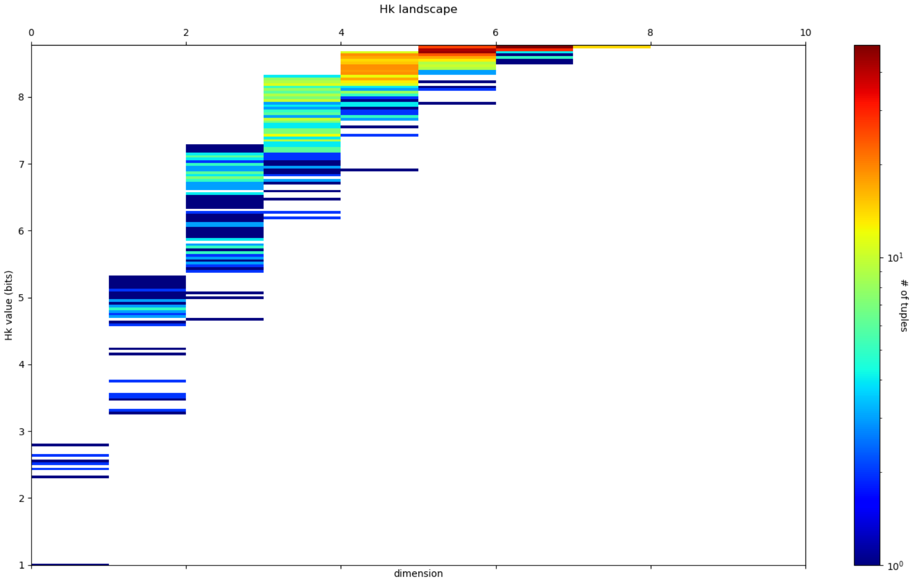
As before, we execute:

```
Nentropie = information_topo.simplicial_entropies_decomposition(iris.data)
information_topo.entropy_simplicial_lanscape(Nentropie)
information_topo = infotopo(dim_to_rank = 4, number_of_max_val = 3)
dico_max, dico_min = information_topo.display_higher_lower_information(Nentropie,␣
↪dataset)
```

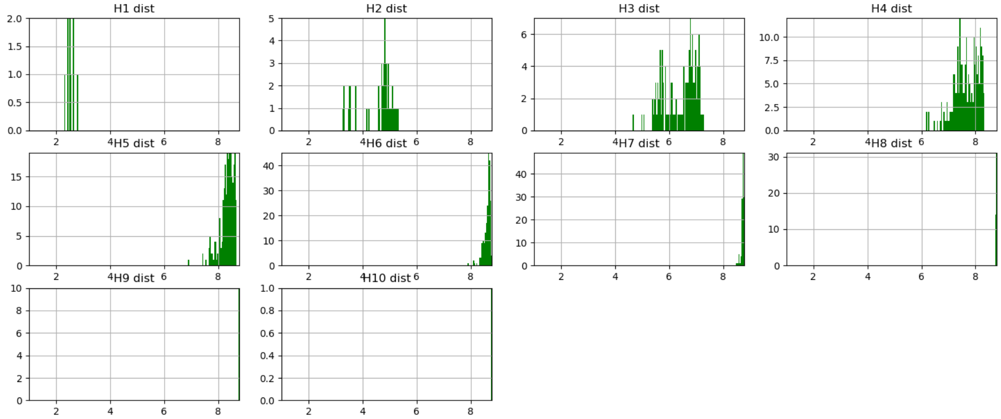
(continues on next page)

(continued from previous page)

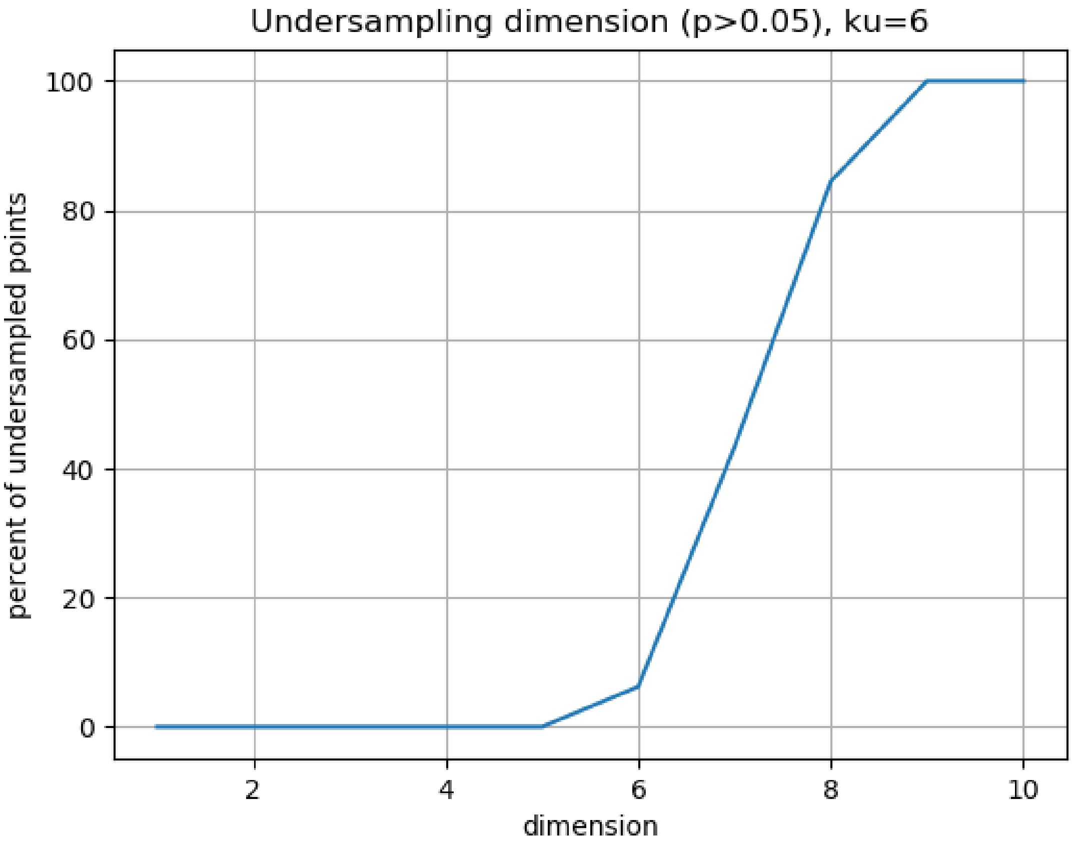
and we obtain the following entropy landscape:



which corresponds to the following distributions of joint entropies for each dimensions:

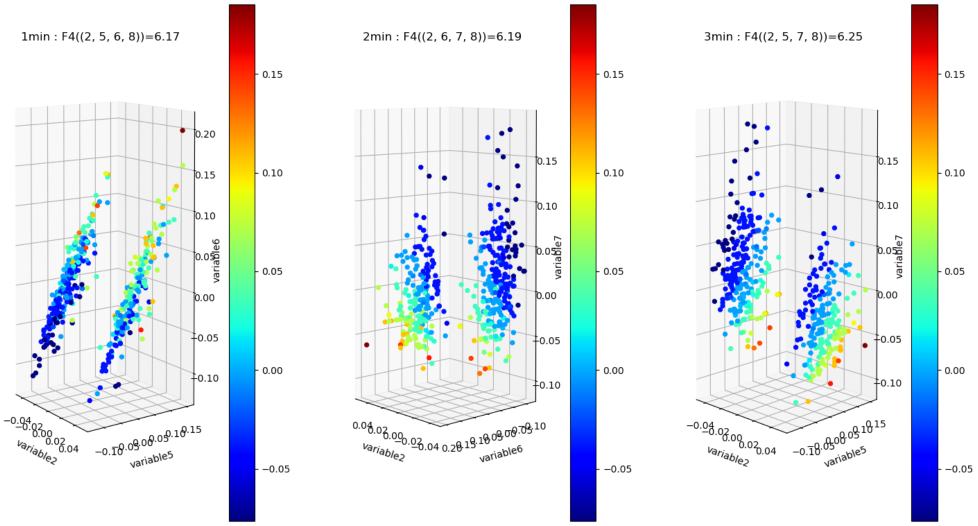


and the computation of the probability of encountering some undersampled probability density estimation (single point box) as a function of the dimension gives:



Which imposing an arbitrary confidence of P>0.05 (default value of the "p_value_undersmapling" parametter), gives a undersampling dimension $k_u = 6$, meaning that with such level of confidence one should not interpret the landscapes and information estimations (whatever) above the 5th dimension. This method is very basic and can (or shall) be improved in several ways, notably a strategy exploring undersampling or information paths should provide more relevant methods, adapted to data structure (to be done).

The number of tuples (a total of 2^{10})) to represent becomes to hudge, and enforces to plot only the distribution histograms of k-tuples value (with a given number of bins = nb_bins_histo) in each dimension. We already see that there exist some interesting structures since the distribution of H_3, H_4, H_5 display obvious bi-modality: the minimum joint entropy mode of the tuples contains the tuples the furthest from randomness. The result shows for example that the 3 first minimum 4-entropy (figure below) contains the binary "sex" variable. It points out one of the current possible limitation-bias of the present algorithm: for heterogeneous variable input, the algorithm should allow different number of values adapted for each variable (binary ternary etc… at the moment their all the same… to be done).

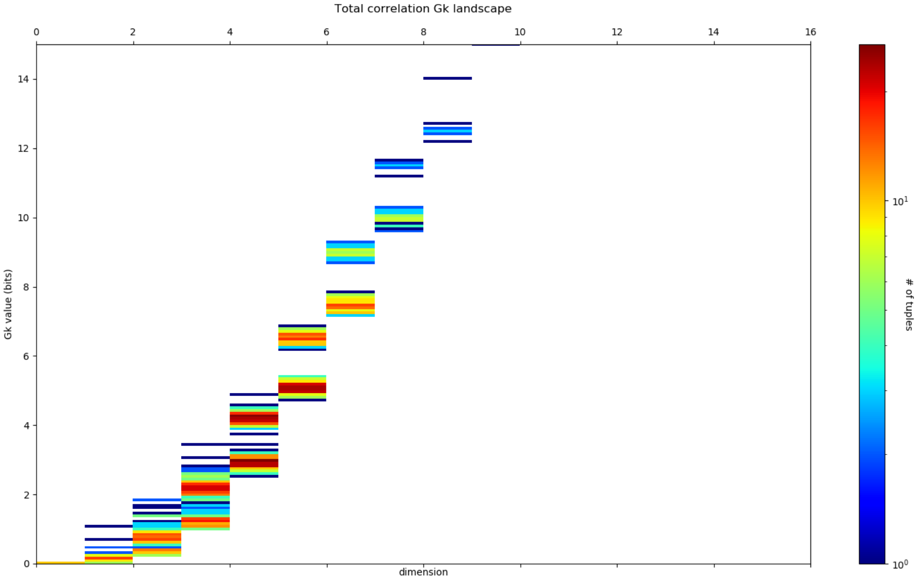


1.2.3 Total correlation

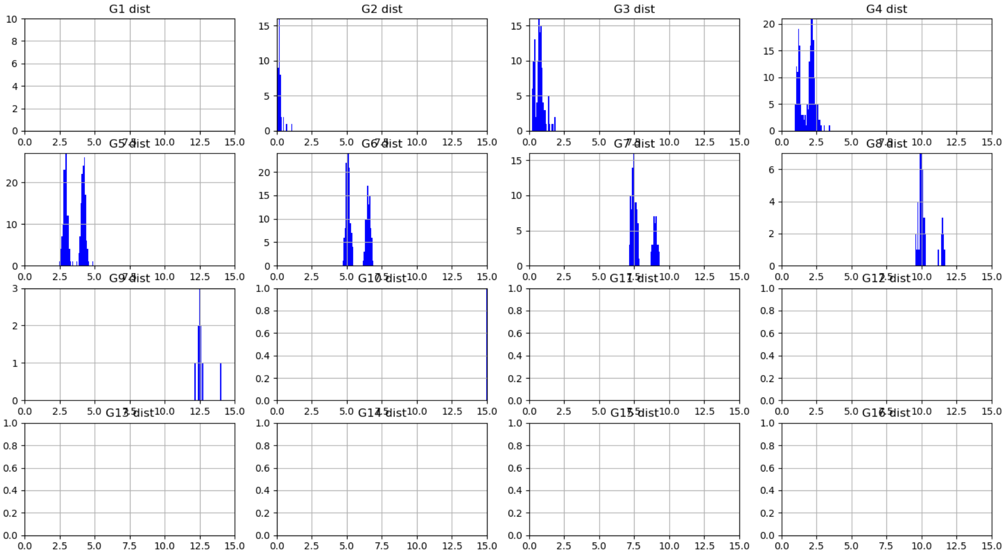
We can now focus on the statistical depencies and G_k and I_k structures, we will first compute the total correlation G_k, by running as previously the commands:

```
Ntotal_correlation = information_topo.total_correlation_simplicial_lanscape(Nentropie)
dico_max, dico_min = information_topo.display_higher_lower_information(Ntotal_
↪correlation, dataset)
```

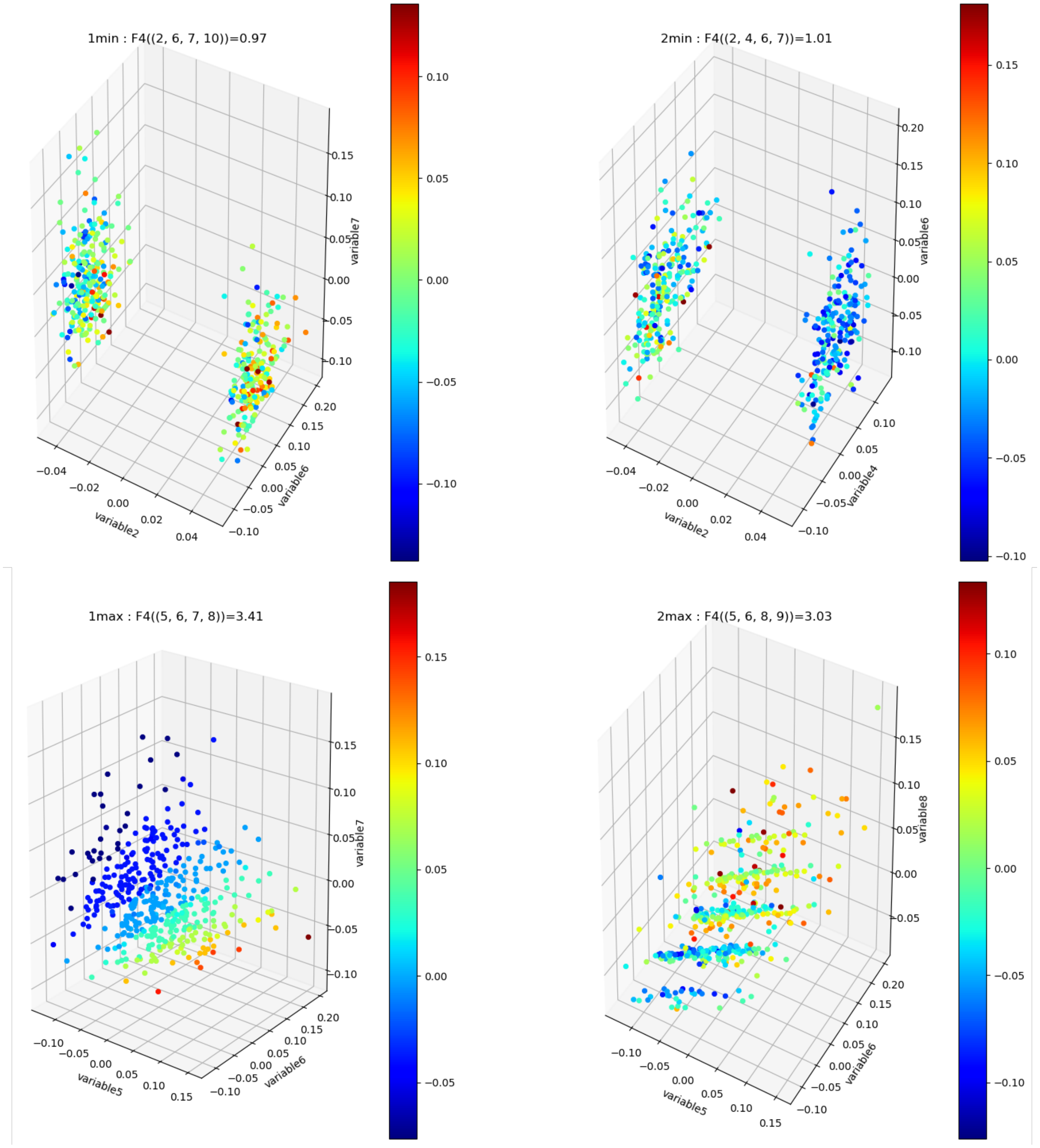
and we obtain the following G_k landscape:



which corresponds to the following distributions of free energy G_k for each dimensions:



The structure of dependences appears much richer, notably the landscape exhibits nice and clearcut bimodal distribution of free energy from dimension 3 to dimension 8. The data points 4-subspace corresponding to the two first minima and maxima of G_4 look like this :



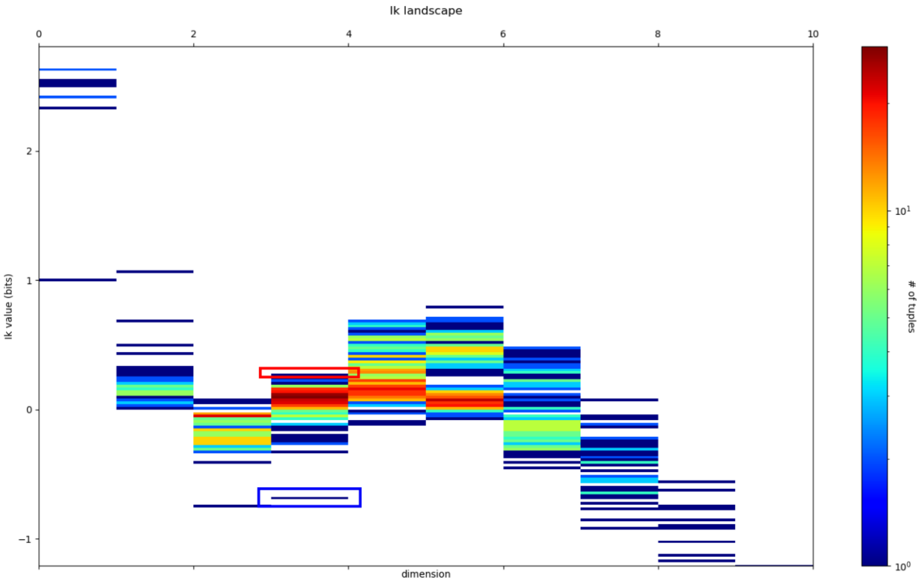
As expected the two G_4 minima present the dependent 4-subspace, but the the two G_4 maxima, for the 4-tuples (5,6,7,8) and (5,6,8,9), present higly dependent very nice statistical dependencies (further detailed in the I_4 subsection bellow).

1.2.4 Mutual Information

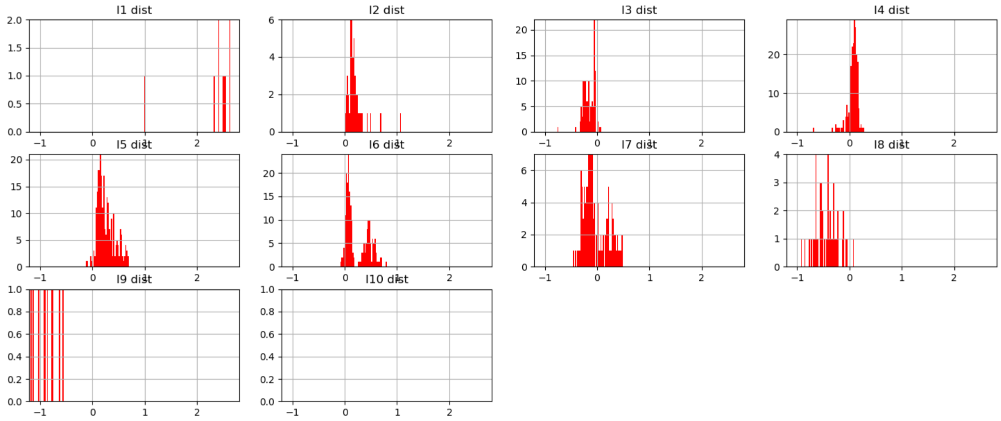
We can now plot similarly the I_k landscape, using the commands:

```
Ninfomut = information_topo.simplicial_infomut_decomposition(Nentropie)
information_topo.mutual_info_simplicial_lanscape(Ninfomut)
dico_max, dico_min = information_topo.display_higher_lower_information(Ninfomut,␣
↪dataset)
adjacency_matrix_mut_info =information_topo.mutual_info_pairwise_network(Ntotal_
↪correlation)
```

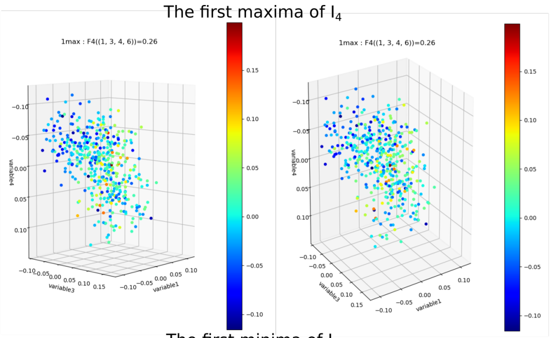
and we obtain the following I_k landscape:

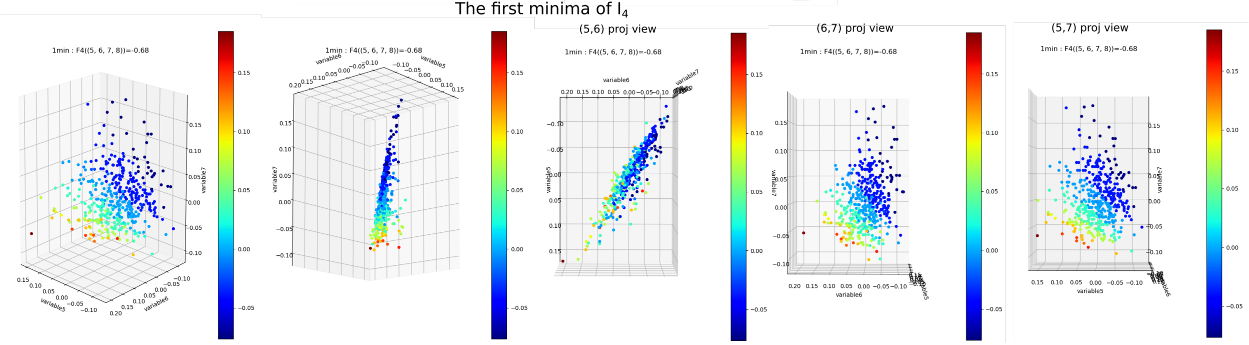


which corresponds to the following distributions of k-mutual information for each dimensions:



I_k landscape bring new results that could not be infered from total correlations, notably thanks to its possible negativity. The I_k landscape of diabetes dataset notably displays important negative values (it was chosen to illustrate this very peculiar phenomena) in dimension 3 and 4 for some 3-tuples and 1 4-tuples (framed in blue). The data points 4-subspace corresponding to this minimal I_4 and the maximal I_4 look like this (with different views) :

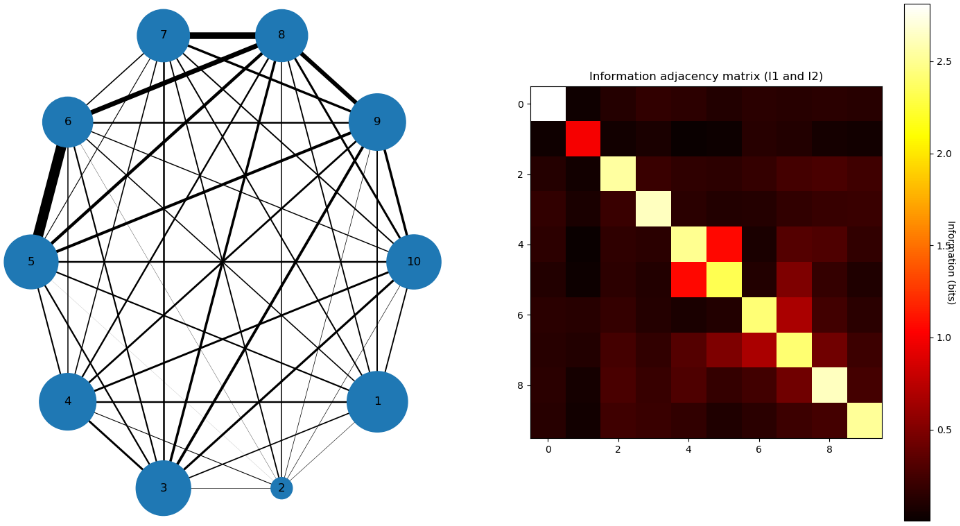




The tuple maximal I_4 (framed in red) only display a weak correlation, as expected from the low I_4 value. However the tuple with minimal I_4 (5,6,7,8) displays an impressive correlation structure taking the form of a 3 dimensional hyperplane (sligtly curved indeed). Looking at projections on 2 dimensional subpaces as shown on the 3 plots on the right we see that the subspace corresponding to the tuples (5,6) and (7,8) is higly "correlated" while (6,7) and (5,7) are highly "random". Indeed, the tuples (5,6), (7,8) and (6,8) obtain the maximum pairwise mutual information. This phenomena of information negativity is known in neuroscience as synergy since the work of Brenner et al. The fact that the 4-tuplet (5,6,7,8) have minimal and not maximal I_4 provides us important additional information that cannot be deduced form the pairwise I_2 (e.g the fact that (5,6) and (7,8) have maximum I_2): the fact that the pair of variables (5,6) and (7,8) and (6,8) untertain causal relationship but have a common cause (another, possibly joint, variable). More precisely we can infer the following causal scheme: $5 \rightarrow 6 \leftrightarrow 8 \leftarrow 7$ (with an ambiguity in the causal dierction between 6 and 8 that could be disambiguated by having a look in the higher dimension 5, and an ambiguity in the global flow, all the arrows could be reversed, that could be desambiguated by looking at lower dimensions). This is indeed equivalent to strong transfer entropy (or conditional mutual information, see Schreiber) but applied here in a general context without time series structure assumption. Transfer entropy is well known to generalize Granger causality to non-linear cases (see Barnet et al). The classical example of a common causal variable is given by: "as ice cream sales increase, the rate of drowning deaths increases sharply.": both are correlated but none of each causes the other. A section in "how_infotopo_works" is dedicated to a more complete study and explanation of these statistical interactions. The gene expression study of Tapia et al. provides further examples of strong positive k-tuplet, e.g of statistical interactions without common cause, or more simply causal chains (e.g metabolic chains). The possiblity to extract causal relation from information structures, I_k landscape, is better illustrated by analysing the LUCAS0 Medical diagnosis dataset sympathicaly proposed by the Causality Challenge #1: Causation and Prediction . It can be acheived by setting the variable dataset_type == 4 in the main of the python script after dowloading the csv at the previous link. In this synthetic training example the 3 variables "smoking", "genetics" and "lung cancer" (1,5,12) are among the minimal I_3 while they were designed to exemplify the causal structure math:*1 rightarrow 12 leftarrow 5*. The dataset and causality results are detailed in the next section.

1.2.5 Information Networks

The information networks representation of I_1 and I_2 for the diabetes dataset is:



The maxima of I_2 are for (5,6) then (7,8) then (6,8) and minima of I_3 are for (5,7,8) then (6,7,8), and this indicate that 5 may cause 7 and 8, and that 6 causes 7 and 8, while 5 and 6 are highly inter-dependent, among other relation, potentially complex relationships that can be infered from the information landscape.

1.2.6 Mean Information path

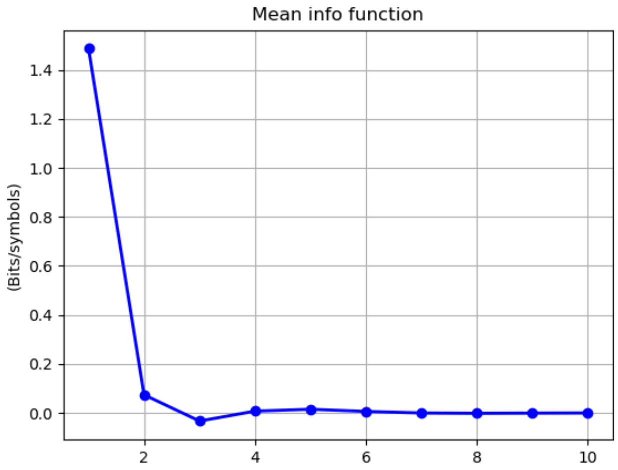
It is interesting to compute and plot the mean I_k paths, which consist in dividing the sum of I_k by the binomial coefficient $\binom{n}{k}$, and the Mean I_k rate , which consist in dividing the preceeding result by the dimension:

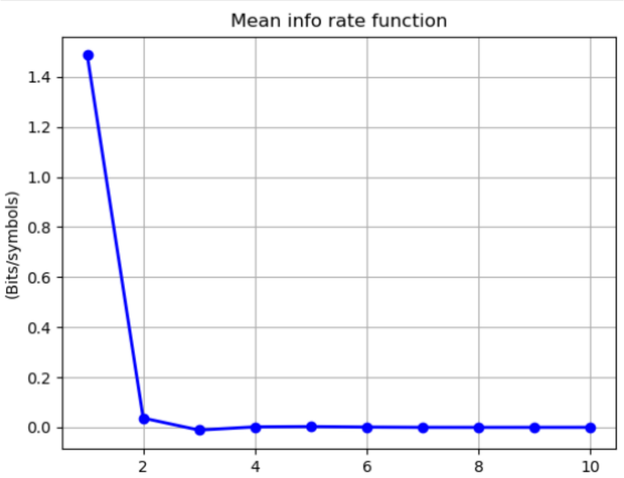
$$\langle I_k \rangle = \frac{\sum_{T\subset[n];card(T)=i} I_k(X_T;P)}{\binom{n}{k}}$$

Using the command:

```
mean_info, mean_info_rate = information_topo.display_mean_information(Ninfomut)
```

we obtain the following mean I_k paths and mean I_k rate paths:





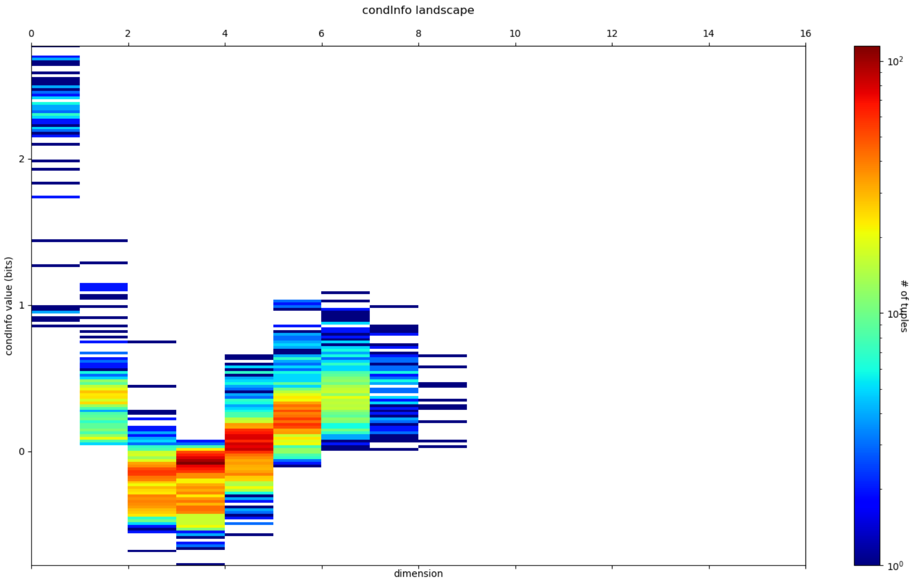
Mean I_k corresponds to the mean-field approxiamtion in statistical physics, that assumes a homogeneous system with identical particles and identical k-body interactions. We recover a usual free-energy landscape analogous to n-bdy van der Waals model, here with a (little) minima at the critical dimension 3, which shows that the interactions (or statistical dependences) in the data are weak in average (almost the independent case). The same computation and definitions can be acheived for k-entropy, and is let as an exercise.

1.2.7 Conditional (transfer) Informations

The visualization of information landscapes as histograms do not permit to visualize and study the conditional entropies and Mutual informations, that can be very interesting as we saw with the (extension) of transfer entropy. They are given by chain rules and correspond to minus the slope of each edges of the lattice in the landscapes. It is possible to plot them using the command:

```
NcondInfo = information_topo.conditional_info_simplicial_lanscape(Ninfomut)
information_topo.display_higher_lower_cond_information(NcondInfo)
```

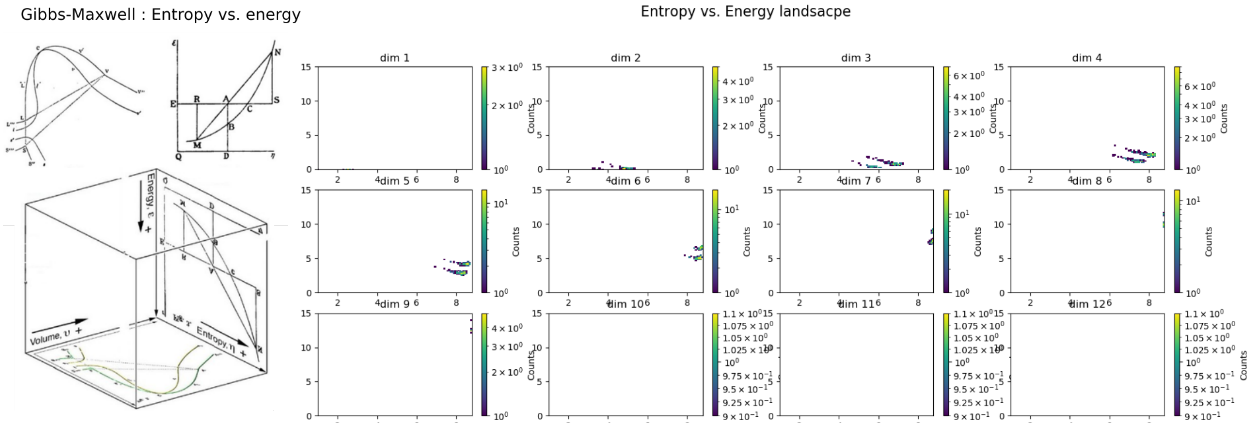
There are more conditional Informations than I_k ($k\binom{n}{k}$ in each k-dimension, and $n2^{n-1}$ in total), and we encoded the output as a list for each dimension, "NcondInfo", of dictionaries which items are of the forms ((5, 7, 9), 0.352) for the information of 5,7 knowing 9, e.g. I(5,7|9). Indeed, as remarked by (Han (1981) Yeung generates all the other information quantities we saw: considering the conditionning variable as the deterministic unit we obtain mutual informations, and considering equivalent variables we obtain conditional entropies and entropies. Both the "Shannonian" and "non-shannonian" inequalities found by Yeung translates directly in information landscapes as bounds on the slope paths (or topological cones), unraveling their homological nature (see PDF). For the diabetes dataset, we obtain:



1.2.8 Entropy Vs. Energy

Following the original figure ENTROPY vs. ENERGY vs. VOLUME of Willard Gibbs (1873) James Clerk Maxwell (1874), we can resume part of the preceding results by ploting H_k (absyssa) vs. G_k (ordinate) using the command:

```
information_topo.display_entropy_energy_landscape(Ntotal_correlation, Nentropie)
```



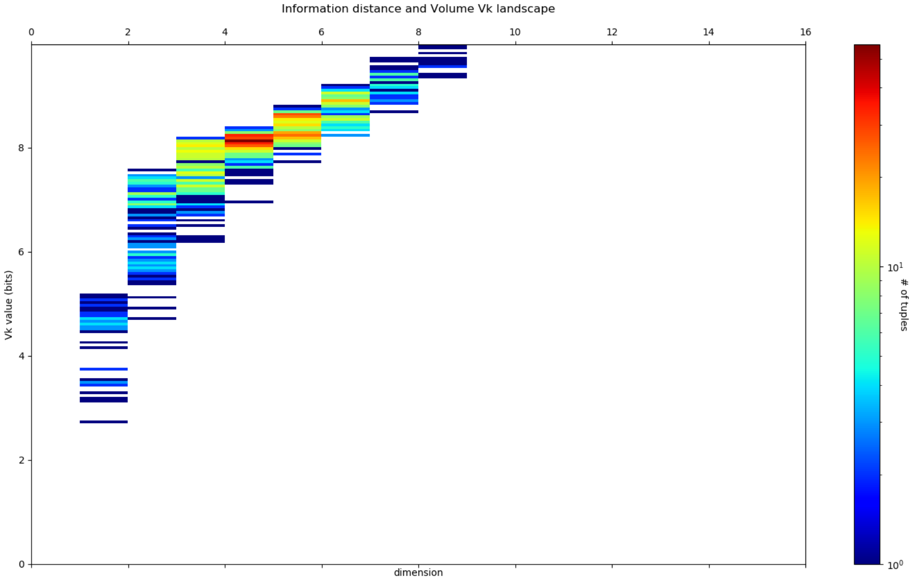
It notably shows how two population of data points clusters from dimension 6 to 8.

1.2.9 Information distance

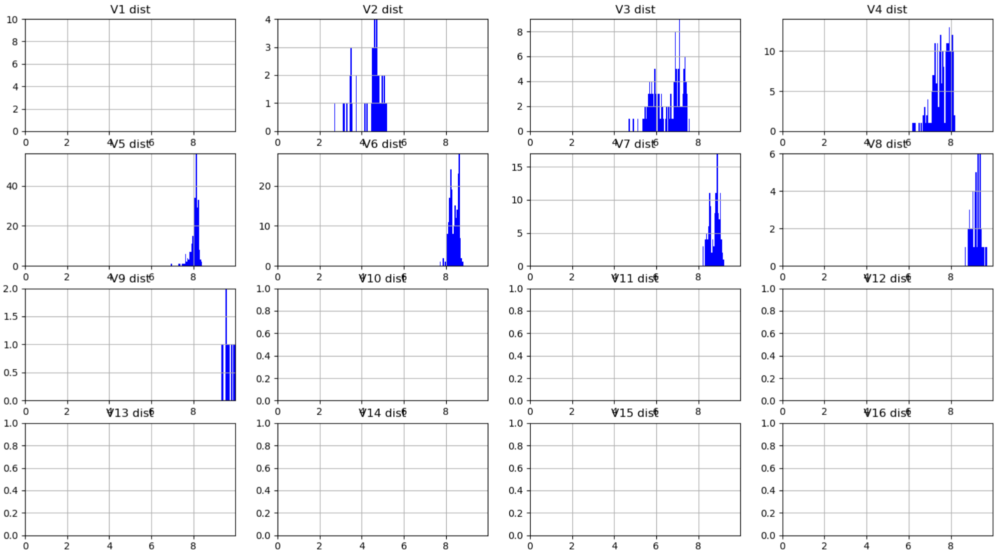
Another nice information measure is information distance or metric defined by $V_2(X;Y) = H_2(X,Y) - I_2(X;Y)$. It is a "real" (and unique see Han for unicity proof metric in the sens that it satifies triangle inequalities and symmetry (precisely except identity if null, it is even better than a metric, it is a pseudo-metric). This metric was find by Shannon (1953) , and was the subject of further interesting both applied and theoretical studies (Han 1981 , Rajski 1961 , Zurek , Bennett and Kraskov and Grassberger). It indeed appears as a topological invariant in a precise setting cohomological setting and generalises to the multivariate case to k information volumes $V_k = H_k(X,Y) - I_k(X;Y)$ PDF . V_k are non-neagtive and symmetric functions. For Machine Learning, this shall be understood as an informational version of Jaccard metric, intersection over union (iou) or other union minus intersection metrics. We can compute their simplicial structure using the commands:

```
Ninfo_volume = information_topo.information_volume_simplicial_lanscape(Nentropie,␣
↪Ninfomut)
dico_max, dico_min = information_topo.display_higher_lower_information(Ninfo_volume,␣
↪dataset)
adjacency_matrix_info_distance = information_topo.mutual_info_pairwise_network(Ninfo_
↪volume)
```

On the Diabete dataset, it gives the following V_k landscape:



with the following distributions:



We see that the structure is less interesting compared to the one we obtained with I_k and G_k, but its geometrical status of a (pseudo)-metric leaves it appealing to plot in its network representation.

Beware that these tools will not detect whatever possible statistical dependencies (see James and Crutchfield PDF), this is just a simplicial subset (nice…paths are automorphism) subsets, computationnally tractable. The complete structure of dependencies are spanned by general information structures and lattice of patition (see section how_infotopo_works), which embedds the present simplicial case. This concludes our introduction to basic infotopo usage – hopefully this has given you the tools to get started for yourself. Further tutorials, covering infotopo parameters and more advanced usage are also available when you wish to dive deeper.(X)

CHAPTER 2

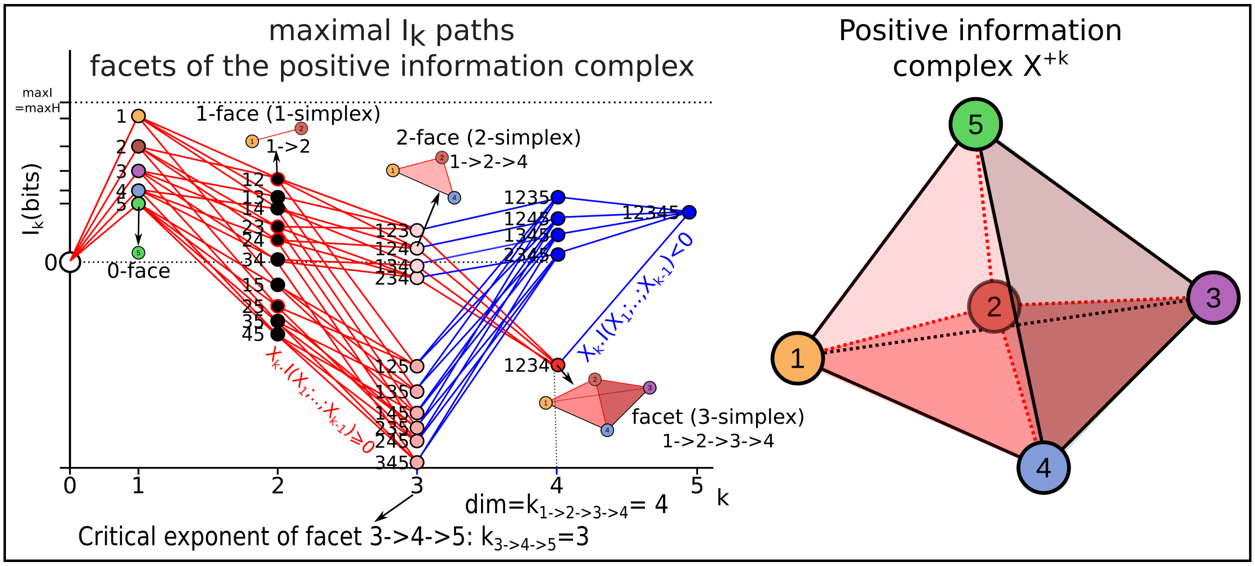
Topological Learning

2.1 Topological Learning principles

2.1.1 Information Complexes

The presentation of the basic methods and principles we made so far mostly relied on basic information lattice decomposition and simplex structure. In what follows, we will go one stepp further by introducing to simplicial complexes of information which can display much richer structures. This will be the occasion to study more in depth information paths, the analog of homotopical paths in information theory. We will consider subcomplex of this simplicial structure, invocating the fact that any simplicial complex can be realized as a subcomplex of a simplex (Steenrod 1947 , p.296).

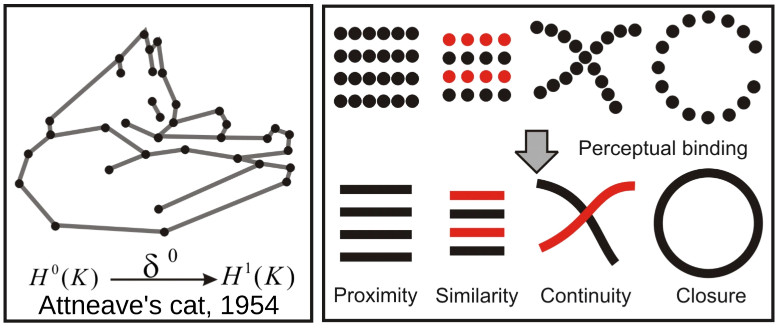
As introduced previously, an information path IP_k of degree k on I_k landscape is defined as a sequence of elements of the lattice that begins at the least element of the lattice (the identity-constant "0"), travels along edges from element to element of increasing degree of the lattice and ends at the greatest element of the lattice of degree k (a piecewise linear function). The first derivative of an IP_k path is minus the conditional mutual information. The critical dimension of an IP_k path is the degree of its first minimum. A positive information path is an information path from 0 to a given I_k corresponding to a given k-tuple of variables such that $I_k < I_{k-1} < ... < I_1$ (a chain, total order). A maximal positive information path is a positive information path of maximal length: it ends at minima of I_k along the path (a minima of the free energy components quantified by I_k. In statistical terms, this minima is equivalent to a conditionnal independence: it means that the conditional mutual information (slope) of the paths goes throug 0. Those maximal paths identifies the maximal faces of the I_k complex and charaterize it, because a simplicial complex is uniquely determined by the list of its maximal faces (intoduction to simplicial homology). Hence, the set of all these paths defines uniquely the I_k complex (or minimum free energy complex). An example of such a complex of dimension 4, with its information path IP_k, is given in this figure:



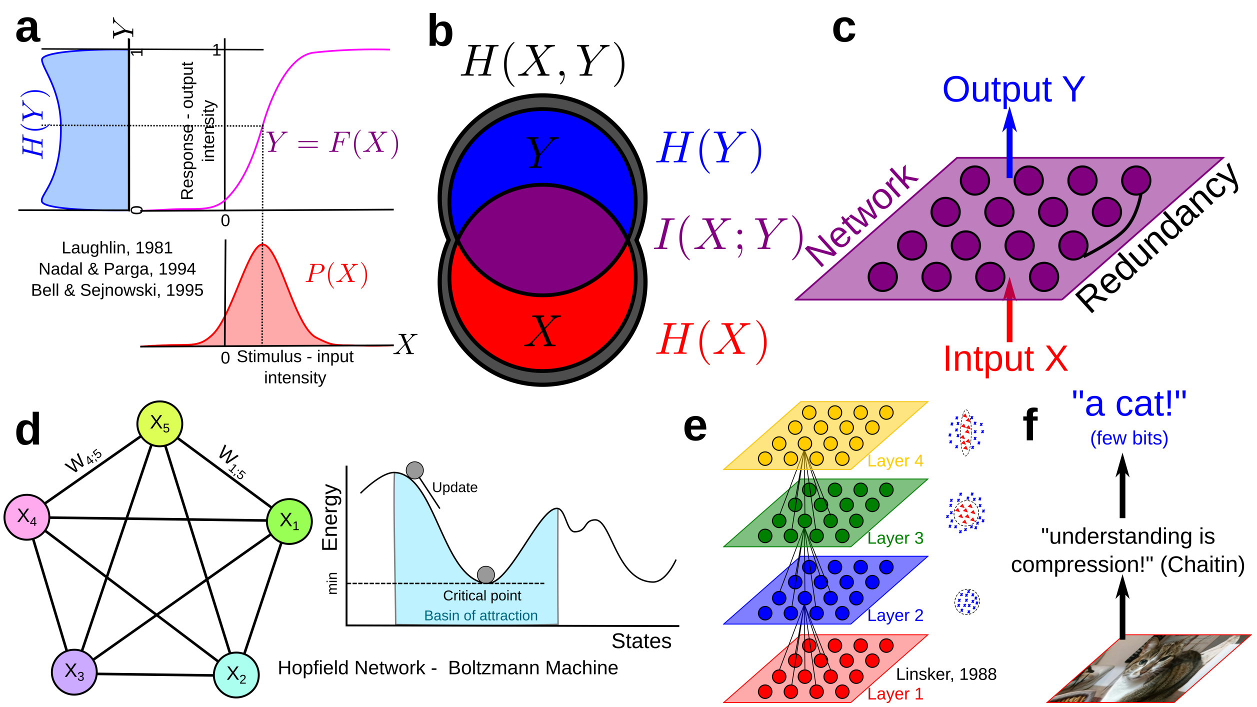
Note that, as a result of classical Shannonian information inequalities, any complex of dimension 3 or below is necessarily a simplex, indicating that in information and statitics, 3 and 4 dimensional topology are also very special.

2.1.2 Poincaré-Shannon Machine

Information theory motivated the early stages of Machine Learning and Information sensory processing theories. The principle was self-resumed by Chaitin: "Understanding is compressing". Notably, Attneave (1954) ennouciated the principles of efficient coding (with Barlow) in the following terms: the goal of sensory perception is to extract the redundancies and to find the most compressed representation of the environment. Any kind of symmetry and invariance are information redundancies and Gestalt principles of perception can be defined on information theoretic terms. This is basically illustrated by, Attneave's famous cat and the topologically sounding Gestalt principle of perceptual binding illustrated bellow:



Since then Information theory has provided machine learning's central functions: the loss functions: Maximum entropy is at the root of Jaynes and may statistical physic inference model, maximum mutual information (infomax) was stated and studied by Linsker, Nadal and Parga, and Bell and Sejnowsky and formalized ICA principles and Hebbian plasticity, generalizing PCA to non-linear cases, Boltzmann Machine minimized the KL-divergence… untill current Deep Convolutional Neural Networks (CNN) that basically minimize cross entropy or "deformed" functions of it like the focal loss (very close indeed to a "deformed probability"!). The principles stayed the same, but Neural network architectures, data availability, computational power and software facilities increased enormously.



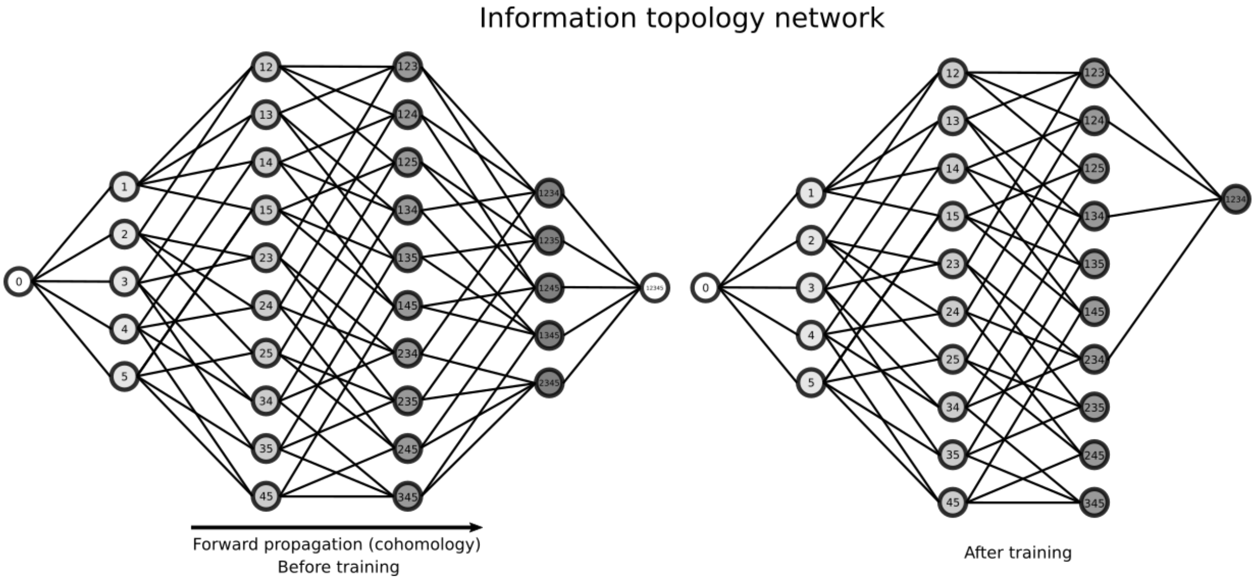
For instance, Boltzmann Machines are reccurent neural networks of binary random variables with hidden layer, that can be formalized as a Markov random field. Markov random fields are a small, positive, subcase of information structures (see proposition 7 (Hu) PDF).

The models developped here are called the Poincaré-Shannon machine in reverence to Boltzmann Machine , Helmholtz Machine and the original Hopfield's network , since it implements simplicial homology (see Poincaré's Analysis Situs , that arguably foundate algebraic topology) and information theory in a single framework (see Shannon's theory of communication , that foundate information theory), applied effectively to empirical data.

The Poincaré-Shannon machine are generic feed forward Deep Neural Networks (DNN) model with a layered structure given by a chain complex (of random variables), e.g. imposed by algebraic topology, and whose connections are given by the edges of the embedding lattice. In the basic simplicial case developped computationnaly here, the rank of the layers of the DNN is the dimension of the faces of the complex, and the highest rank of the layers is the dimension of the complex. As with usual DNN, the dimension of the analized patterns increases with the depth of the layers (see for illustration, figure 2 Zeiler and Fergus 2013) The neurons are random variables, and are whatever measurable functions (linear, non linear), hence covering a "fairly" large class of functions (notably, using the Solovay's axiomatic of set theory, all functions are measurable). In the general (and computationally hard) setting of general information strutures, that considers the lattice of partitions (cf. section "how infotopo works"), the Poincaré-Shannon machine are Universal Classifiers, in the sens that a partition corresponds exactly to an equivalence class and in theory such a model would span all classifications up to equivalence). This topological structure allows, and de facto implements the fact, that neural layers are not necessarilly serial as in current DNN, but can be parralel. Such architectures are well known in real sensory cortical systems, for example the ventral and dorsal visual streams in human cortex would corresponds to two facets of the human brain complex with two (at least partially disjoint information paths) and analyze conditionally independent features of the input such as the "where and what" (dorso and ventral, respectively PDF). Hence one of the interest of such deep model is that its architecture (number of layers, number of neurons at each layer, connectivity) and computation is fully mastered and understood, as far as simplicial complexes can be: it can be understood as an algebrization of neural networks (there are other very interesting approachs of such topic (not probabilistic as here), see for example the neural rings (Curto and Youngs 2013-2020) or Morisson et al. 2016 Morisson and Curto 2018).

Beside this architectural difference with usual DNN, the second important difference is that the learning rule is a "forward propagation", imposed by the cohomological "direction", whereas usual DNN implements a backpropagation learning rule (homological "direction") which implements basically the chain rule of derivation (Kelley 1960 , Le Cun

1985, Dreyfus 1962, Rumelhart et al. 1986). The information topology take profit of the coboundary nature of I_k functions, a (discrete in the present particular case) statistical analog of differential operator. This means that there is no descent as in the usual DNN implementation, but that computation of those I_k and conditional I_k implements the descent. Notably, the introduction of the multiplicity decomposition of "energy functions" formalizes learning in neural networks in terms of a combinatorial family of analytically independent functions I_k (moreover with independent gradients) on the probability simplex (Han 1975 Han 1978 Theorem 4 in PDF): instead of a single energy and associated gradient descent, mutual information provides a multiplicity of gradients. The following illustration presents the DNN architecture associated with the previous example of an I_4 complex:



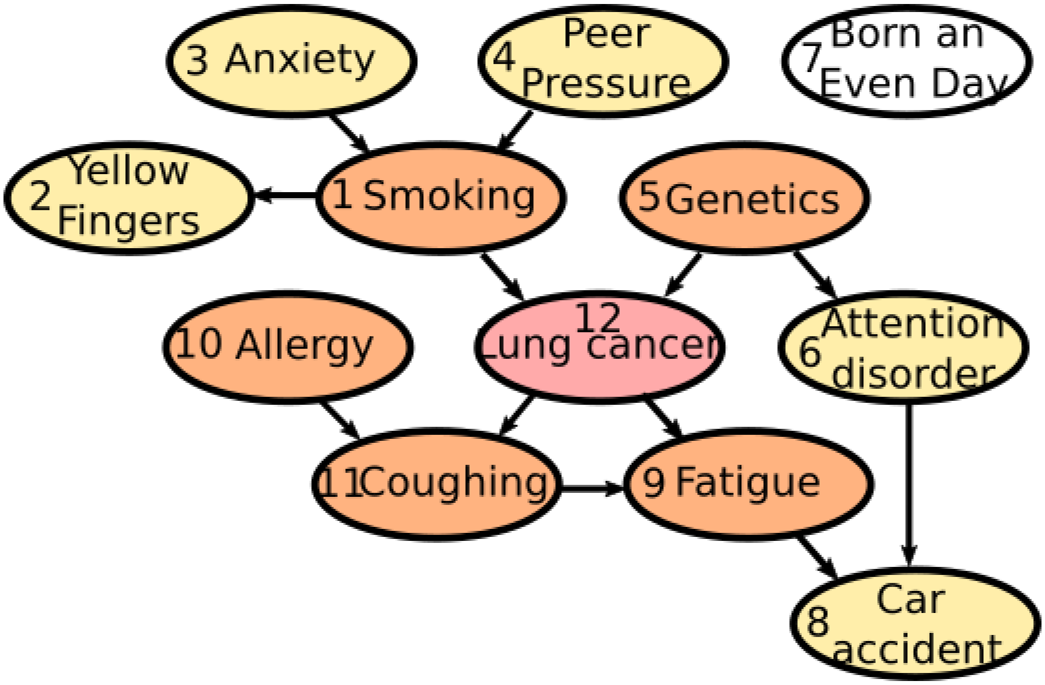
2.2 Unsupervised topological learning

2.2.1 Causality challenge dataset

We will illustrate the computation of free energy complex (or I_k complex) on the synthetic dataset LUCAS (LUng CAncer Simple set) of the causality challenge. Before trying the code on your computer, you will have to download the file "lucas0_train.csv" and to save it on your hard disk (here at the path "/home/pierre/Documents/Data/lucas0_train.csv"), and to put your own path in the following commands with the initialisation of infotopo's parameters.

```
import pandas as pd
dataset = pd.read_csv(r"/home/pierre/Documents/Data/lucas0_train.csv")   # csv to␣
↪download at http://www.causality.inf.ethz.ch/data/LUCAS.html
dataset_df = pd.DataFrame(dataset, columns = dataset.columns)
dataset = dataset.to_numpy()
information_topo = infotopo(dimension_max = dataset.shape[1],
                            dimension_tot = dataset.shape[1],
                            sample_size = dataset.shape[0],
                            work_on_transpose = False,
                            nb_of_values = 2,
                            sampling_mode = 1,
                            deformed_probability_mode = False,
                            supervised_mode = False,
                            forward_computation_mode = False,
                            dim_to_rank = 3, number_of_max_val = 4)
```

The dataset is composed of 11 variables: 1: Smoking, 2: Yellow_Fingers, 3: Anxiety, 4: Peer_Pressure, 5: Genetics, 6: Attention_Disorder, 7: Born_an_Even_Day, 8: Car_Accident, 9: Fatigue, 10: Allergy, 11: Coughing and the 12th variable of iterest: Lung cancer. The (buildin) causality chain relations among those varaibles follow this schema:

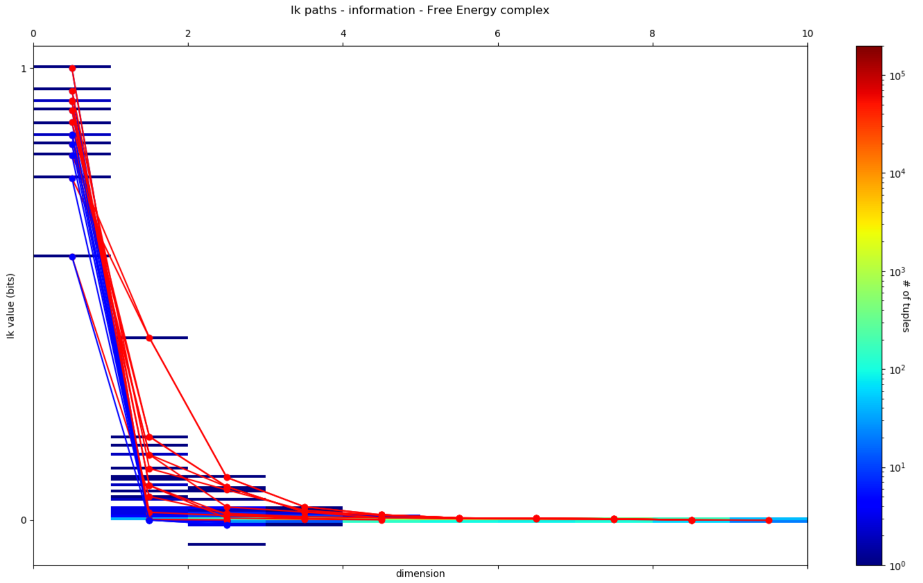


2.2.2 Information Complexes

To compute (approximation) of the information complex (free-energy complex), you can use the following command:

```
Ninfomut, Nentropie = information_topo.fit(dataset)
information_topo.information_complex(Ninfomut)
```

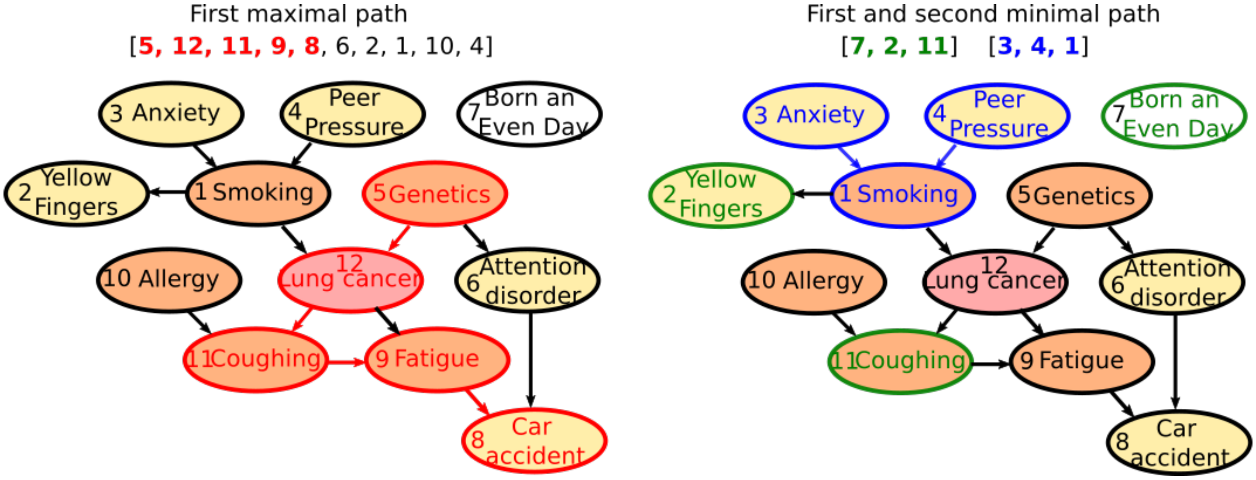
The method “fit” is just a wrapper of the methods “simplicial_entropies_decomposition” and “simplicial_infomut_decomposition”, that is introduced to correspond to the usual methods of scikit-learn, keras, tensorflow (…). The set of all paths of degree-dimension k is intractable computationally (complexity in $\mathcal{O}(k!)$). In order to bypass this issue, the current method “information_complex” computes a fast local algorithm that selects at each element of degree k of a path, the positive information path with maximal or minimal I_{k+1} value (equivalently, extremal conditional mutual informations) or stops whenever $X_k.I_{k+1} \leq 0$ and ranks those paths by their length. The justification of this elementary heuristic is that it should capture the paths with the most interesting tuples, e.g the one highest anf lowest I_k. No doubt that this approximation is rought and shall be improved in future (to be done). The result on the causality challenge dataset is:



and it prints the following paths:

```
The path of maximal mutual-info Nb 1  is : [5, 12, 11, 9, 8, 6, 2, 1, 10, 4], The path of minimal mutual-info Nb 1  is : [7, 2, 11], The path of maximal mutual-info Nb 2  is :[2, 12, 11, 9, 3, 6, 10, 5], The path of minimal mutual-info Nb 2  is : [3, 4, 1], The path of maximal mutual-info Nb 3  is : [1, 2, 12, 11, 9, 3, 6, 10, 5], The path of minimal mutual-info Nb 3  is : [10, 4, 7], The path of maximal mutual-info Nb 4  is : [9, 11, 12, 1, 2, 3, 6, 10, 5], The path of minimal mutual-info Nb 4  is : [4, 3, 1], The path of maximal mutual-info Nb 5  is :[8, 9, 11, 12, 5, 6, 2, 1, 10, 4], The path of minimal mutual-info Nb 5  is : [6, 1, 12] etc..
```

The first maximal path [5, 12, 11, 9, 8, 6, 2, 1, 10, 4] as length 10 and the first 5 variables corresponds to one of the longest causal chain of the data as illustrated bellow. The fact that the resulting path is so long is likely due to the generating algorithm used for Lucas, and the last [6,2,1,10,4] errors could be removed by statistical test thresholding on conditional mutual information values. The next maximal paths fail to identify the other long causal chain of the data, probably as a consequence of the rought approximation used by the algorithm. The First two minimal paths [7, 2, 11] and [3, 4, 1] identifies unrelated variables or multiple cause causality scheme.



Beware, that the computational heuristic provided does not give a chain complex in the algebraic topology sens, but just a partial approximate view: notably due to the heuristic some chains identified by the algorithm may included in other (and hence may bot be maximal faces). The excat computation of the information is acheived by running this code:

```
N_info_paths = information_topo.information_paths(Ninfomut)
```

Its output are the maximal faces of the complex. However, at least with the current code and a basic computer, it is hopeless to run it for problems with dimension above 10.

2.2.3 Digits Dataset

In order to illustrate unsupervised and supervised learning methods we will now turn to the classical dataset of Digits NIST, which is a common toy dataset to train and test machine learning models. We load it as previously using the symapthic Scikit-learn repository:

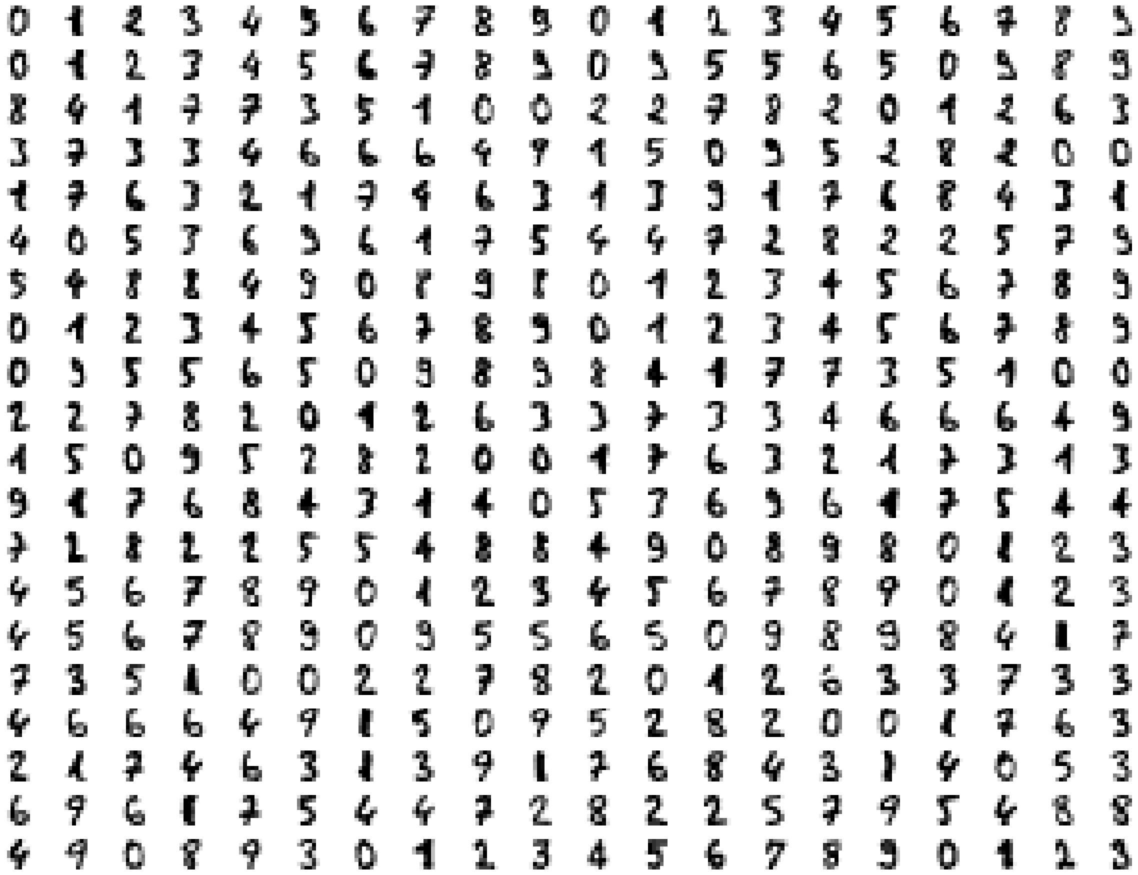
```
dataset = load_digits()
print(dataset.DESCR)
fig, ax_array = plt.subplots(20, 20)
axes = ax_array.flatten()
for i, ax in enumerate(axes):
    ax.imshow(dataset.images[i], cmap='gray_r')
plt.setp(axes, xticks=[], yticks=[], frame_on=False)
plt.tight_layout(h_pad=0.5, w_pad=0.01)
dataset_df = pd.DataFrame(dataset.data, columns = dataset.feature_names)
dataset_df = pd.DataFrame(dataset.data, columns=dataset.feature_names)
dataset = dataset.data
```

It prints the following complete description of the dataset:

```
Optical recognition of handwritten digits dataset
--------------------------------------------------

Data Set Characteristics:

:Number of Instances: 5620
:Number of Attributes: 64
:Attribute Information: 8x8 image of integer pixels in the range 0..16.
:Missing Attribute Values: None
:Creator: E. Alpaydin (alpaydin '@' boun.edu.tr)
:Date: July; 1998

This is a copy of the test set of the UCI ML hand-written digits datasets ␣
↪https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+
Digits

The data set contains images of hand-written digits: 10 classes where each␣
↪class refers to a digit.

Preprocessing programs made available by NIST were used to extract normalized␣
↪bitmaps of handwritten digits from a preprinted form. From a
total of 43 people, 30 contributed to the training set and different 13 to␣
↪the test set. 32x32 bitmaps are divided into nonoverlapping blocks of
4x4 and the number of on pixels are counted in each block. This generates an␣
↪input matrix of 8x8 where each element is an integer in the range
```

```
0..16. This reduces dimensionality and gives invariance to small distortions.

For info on NIST preprocessing routines, see M. D. Garris, J. L. Blue, G., T.␣
↪Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, and C.
L. Wilson, NIST Form-Based Handprint Recognition System, NISTIR 5469, 1994.

.. topic:: References

  - C. Kaynak (1995) Methods of Combining Multiple Classifiers and Their␣
↪Applications to Handwritten Digit Recognition, MSc Thesis, Institute of
    Graduate Studies in Science and Engineering, Bogazici University.
  - E. Alpaydin, C. Kaynak (1998) Cascading Classifiers, Kybernetika.
  - Ken Tang and Ponnuthurai N. Suganthan and Xi Yao and A. Kai Qin. Linear␣
↪dimensionalityreduction using relevance weighted LDA. School of
    Electrical and Electronic Engineering Nanyang Technological University.␣
↪2005.
  - Claudio Gentile. A New Approximate Maximal Margin Classification␣
↪Algorithm. NIPS. 2000.
```

And illustrates the dataset with the following sample of digits pictures:



2.2.4 Adaptive computational complexity

The images of digits dataset are 8*8 pixels, meaning that we have 64 Random Variables or dimensions: this will introduce us to the problemantic of high dimensional space (here not so high) and of computational complexity. In this case the information simplicial structure has 2^{64} information estimations to compute, which is much too big, and we propose a partial exploration that will stop the computation at a given dimension "dimension_max". This methods of partial exploration allows to adapt the computational (time) complexity of the algorithm to a reasonable complexity given your computational ressources and the dimension of the dataset. As we have seen, when increasing the dimension of the dataset, the raw computation potentially grows as $\mathcal{O}(2^n)$. In order to master and circumvince this problem, a partial exploration of information structures as been written, allowing to explore only all the k first dimensions with $n \geq k$. This is acheived by setting the parametter "dimension_max" to k and "forward_computation_mode" to "True". For example, setting "dimension_max=2" will restrict the computation to the $\binom{n}{1} = n$ and the $\binom{n}{2} = n!/(2!(n-2)!) = n.(n-1)/2$ estimations of information, which is the (symetric) usual complexity $\mathcal{O}(n^2)$ of metric or graph based machine learning algorithm. Setting to 3, there will be $\binom{n}{1} + \binom{n}{2} + \binom{n}{3}$ estimations of information giving a complexity in $\mathcal{O}(n^3)$ etc… Of course, we gain what we loose, and the deployement of infotopo on GPU should give a bit more of ressources (currently failed). In 64 dimensions, choosing an exploration of the 5 first dimensions (64+2016+41664+635376+7624512=8303632 estimations) gives a reasonably long computation of several hours on a personal laptop (has acheived here) To set such exploration, we initialize infotopo using the commands:

```
information_topo = infotopo(dimension_max = 5,
                            dimension_tot = dataset.shape[1],
                            sample_size = dataset.shape[0],
                            work_on_transpose = False,
                            nb_of_values = 17,
                            sampling_mode = 1,
                            deformed_probability_mode = False,
                            supervised_mode = False,
                            forward_computation_mode = True,
                            dim_to_rank = 3, number_of_max_val = 4)
```

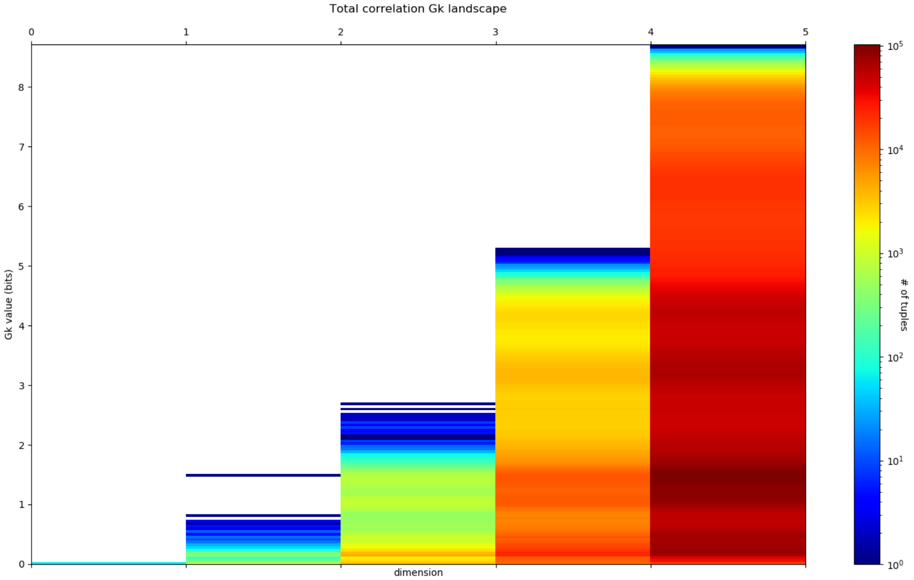
For data scientist used to deep learning terminology, this intialization corresponds to building the model, although extremely simple. As you see, the whole structure of the model is fully constrained by the dataset's embedding dimension, the dimension max (computational complexity restriction), and the number of values chosen for the variables (with other purely computational internal parameter).

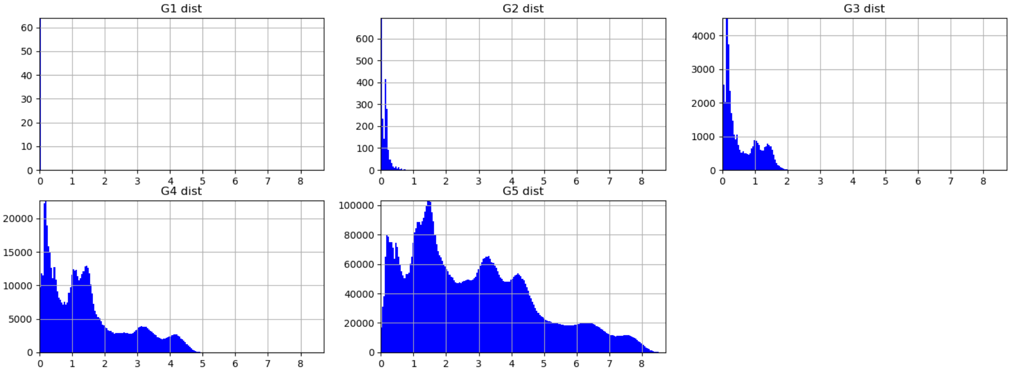
2.2.5 Natural image statistics

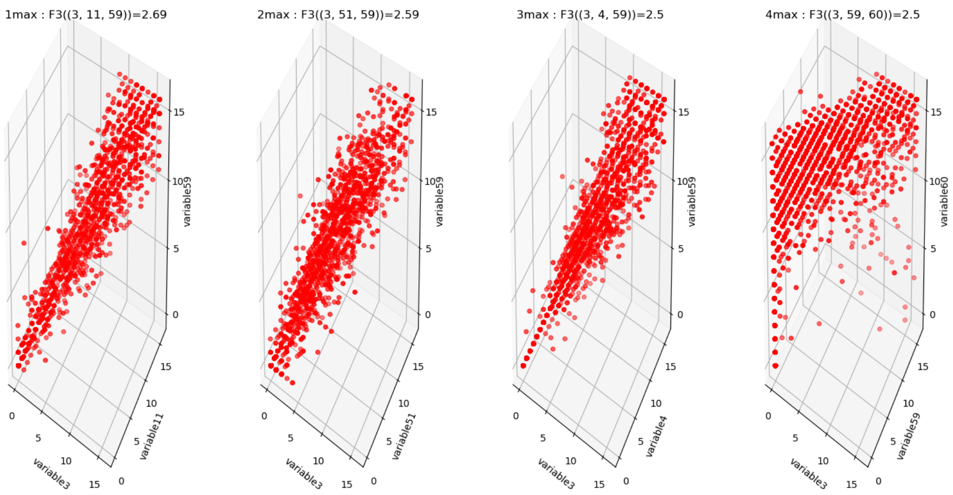
We now apply the model to digits dataset in a first unsupervised case that considers the pixels as variables-dimension up the fifth dimension. It means that we will consider all statistical dependencies, or statistically dependent patterns composed of up to 5 pixels in the digits datset. In computational vision and neursocience, such a task pertains to the domain of natural image statistics studies (see notably chap. 1.4 of the book "natural image statistics", Hyvärinen et al 2009). Of course, the example will present here only a very small subset of natural image statistics corresponding to human hand written digits, but the principle of the study stays the same for other kind of images. We fit the model and display the informations landscape by running the following code:

```
Ninfomut, Nentropie =  information_topo.fit(dataset)
Ntotal_correlation = information_topo.total_correlation_simplicial_lanscape(Nentropie)
dico_max, dico_min = information_topo.display_higher_lower_information(Ntotal_
↪correlation, dataset)
information_topo.mutual_info_simplicial_lanscape(Ninfomut)
dico_max, dico_min = information_topo.display_higher_lower_information(Ninfomut,␣
↪dataset)
```

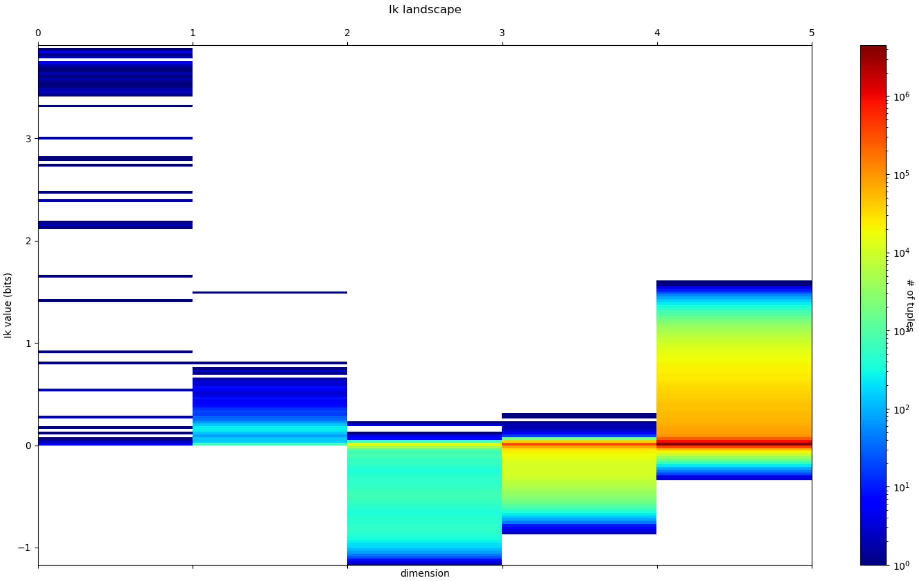
The output of fit gives the model of the data, it concentrates the computationaly hard tasks, and just as it is usually acheived with Neural Network training, for large dimensional problems, it is recommended to save the model (Nentropy,Ninfomut) in a file (using for example pickle, a saving procedure in hdf5 will be written, to be done). The free energy or total correlation G_k landscape, and its 4 maxima triplets data subspace we obtain are the following:

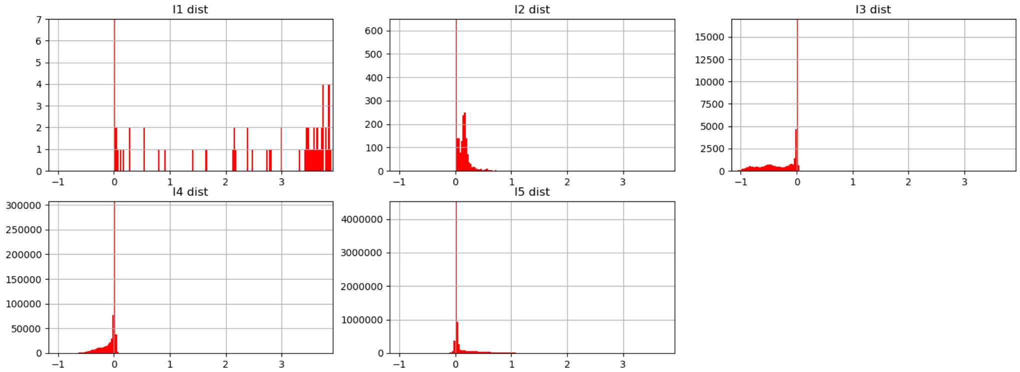


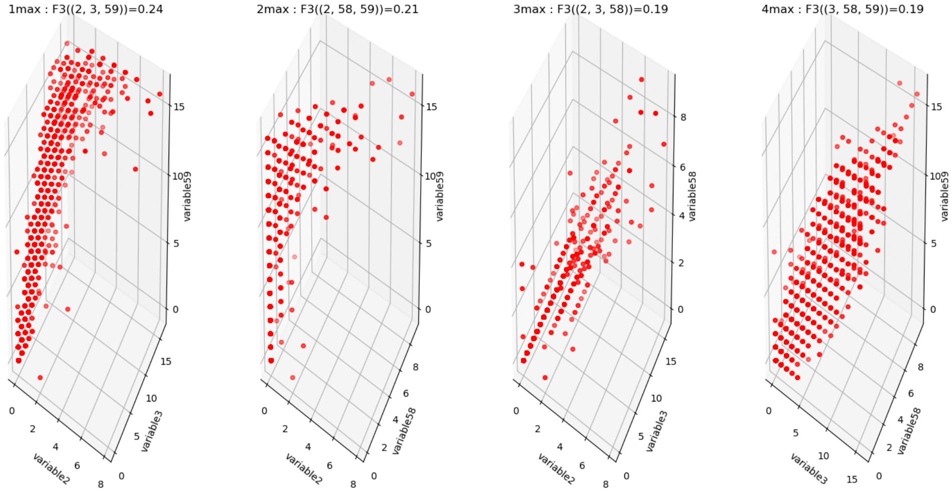




The mutual information I_k landscape, and its 4 maxima triplets data subspace we obtain are the following:







Both landscapes shows the presence of important higher order statitical patterns, with clear and complex multimodal distributions. For those who where not convinced yet that higher orders statictics matters, it now shall be the case: they are indeed the support of our (higher level) every day world's perception and object recognition. I_k landscape notably shows that most k-uplets of pixels are k-independent (e.g. $I_k = 0$). The computation of information distances and volume, joint entropies, conditional informations (. . .) are left as an exercise, but all of them present some meaningfull distributions-spectra.

2.2.6 Convolutional methods

The preceding computation of statistical structures in images has the default of not being translation invariant, made obvious here by the centering preprocessing of the digits in the images. Such a potential problem is easily overcome by using basic convolutional patches instead of the direcy images, just as Convolutional layer of Neural Network do. Note that implementing an information network corresponding to current Convolutional Network would require an iterative process of pooling the maximal information modules and of landscape info paths computation (to be done both theoretically and in practice). However, such a method appears more for the moment as a computational trick to reduce computation rather than a firmly theoretically established method. To extract convolutional patchs of the images we use the function "convolutional_patchs". The function extracts images patchs of m*m pixels (with $m = \lfloor \sqrt{dimension_max} \rfloor$) by sliding on the images. As the function change the matrix of data input and its shape, the function reset automatically dimension_tot=dimansion_max to $(\lfloor \sqrt{dimension_max} \rfloor)^2$ and the sample_size to $sample_size.(n-(m-1))^2$ (where image are n*n pixels and patchs are m*m pixels, and there are $(n-(m-1))^2$ convolutional patchs in a single image). In the example below we set dimansion_max to 16 and hence patchs of 4*4 pixels in sample_size=100 digits images, and we obtain 2500 points-patchs in 16 dimensions.

2.3 Supervised topological learning

CHAPTER 3

Parameters

3.1 Algorithmic parameters

3.1.1 dimension_max

(integer) maximum Nb of Random Variable (column or dimension) for the exploration of the cohomology and lattice

3.1.2 dimension_tot

(integer) total Nb of Random Variable (column or dimension) to consider in the input matrix for analysis (the first columns)

3.1.3 sample_size

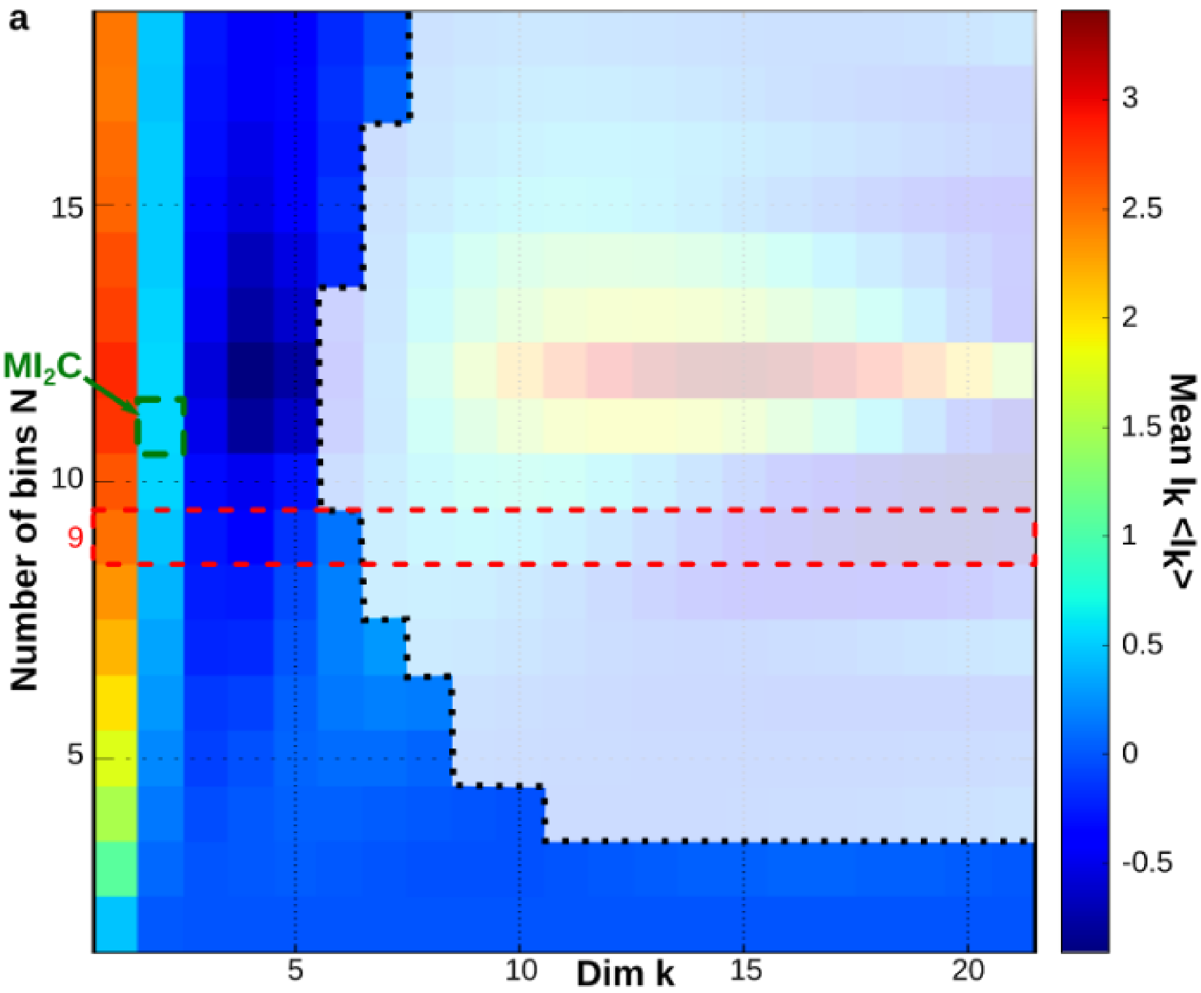
(integer) total Nb of points (rows or number of trials) to consider in the input matrix for analysis (the first rows)

3.1.4 work_on_transpose

(Boolean) if True take the transpose of the input matrix (this change column into rows etc.)

3.1.5 nb_of_values

(integer) Number of different values for the sampling of each variable (alphabet size). It is recommanded to tune this parametter in case of continuous variable. A too small number of values will result in a biased estimation of informations functions that homogenizes information structures, while a too large number of values with respect to the sample size will lead to a fast curse of dimensionality and an interpretation restricted to low dimensions as depited in the following figure. The method proposed to tunne nb_of_values is adapted from Maximum Mutual Information Coefficient (MIC, Reshef et al. 2011) and is describded in section 6.6 PDF . It consist in evaluating the mean mutual information path for a set of different values of nb_of_values together with the undersampling dimension k_u and to choose the one that give the maximum value of mean I_2, as depicted bellow, or a compromise between this maximum value and a not to low k_u. An automatic tuning preocedure could be written (to be done).



3.1.6 sampling_mode

(integer: 1,2,3)

_ sampling_mode = 1: normalization taking the max and min of each columns (normaization row by columns)

_ sampling_mode = 2: normalization taking the max and min of the whole matrix

3.1.7 deformed_probability_mode

(Boolean) The method associated to true do not work yet (to be done)

_ deformed_probability_mode = True : it will compute the "escort distribution" also called the "deformed probabilities". *p(n,k)= p(k)^n/ (sum(i)p(i)^n , where n is the sample size.

[1] Umarov, S., Tsallis C. and Steinberg S., On a q-Central Limit Theorem Consistent with Nonextensive Statistical Mechanics, Milan j. math. 76 (2008), 307–328

[2] Bercher, Escort entropies and divergences and related canonical distribution. Physics Letters A Volume 375, Issue 33, 1 August 2011, Pages 2969-2973

[3] A. Chhabra, R. V. Jensen, Direct determination of the f(α) singularity spectrum. Phys. Rev. Lett. 62 (1989) 1327.

[4] C. Beck, F. Schloegl, Thermodynamics of Chaotic Systems, Cambridge University Press, 1993.

[5] Zhang, Z., Generalized Mutual Information. July 11, 2019

_ deformed_probability_mode = False : it will compute the classical probability, e.g. the ratio of empirical frequencies over total number of observation

[6] Kolmogorov 1933 foundations of probability

3.1.8 supervised_mode

(Boolean) if True it will consider the label vector for supervised learning; if False unsupervised mode

3.1.9 forward_computation_mode

(Boolean)

_ forward_computation_mode = True : it will compute joint entropies on the simplicial lattice from low dimension to high dimension (co-homological way). For each element of the lattice of random-variable the corresponding joint probability is estimated. This allows to explore only the first low dimensions-rank of the lattice, up to dimension_max (in dimension_tot)

_ forward_computation_mode = False : it will compute joint entropies on whole simplicial lattice from high dimension to the marginals (homological way). The joint probability corresponding to all variable is first estimated and then projected on lower dimensions using conditional rule. This explore the whole lattice, and imposes dimension_max = dimension_tot

3.1.10 p_value_undersampling

(real in]0,1[) value of the probability that a box have a single point (e.g. undersampled minimum atomic probability = 1/number of points) over all boxes at a given dimension. It provides a confidence to estimate the undersampling dimenesion Ku above which information etimations shall not be considered.

3.1.11 compute_shuffle

(Boolean)

_ compute_shuffle = True : it will compute the statictical test of significance of the dependencies (pethel et hah 2014) and make shuffles that preserve the marginal but the destroys the mutual informations

_ compute_shuffle = False : no shuffles and test of the mutual information estimations is acheived

3.1.12 p_value

(real in]0,1[) p value of the test of significance of the dependencies estimated by mutual info the H0 hypotheis is the mutual Info distribution does not differ from the distribution of MI with shuffled higher order dependencies

3.1.13 nb_of_shuffle

(integer) number of shuffles computed

3.2 Display parameters

3.2.1 nb_bins_histo

(integer) number of values used for entropy and mutual information distribution histograms and landscapes.

3.2.2 dim_to_rank

(integer) chosen dimension k to rank the k-tuples as a function information functions values.

3.2.3 number_of_max_val

(integer) number of the first k-tuples with maximum or minimum value to retrieve in a dictionary and to plot the corresponding

points k-subspace.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

CHAPTER 5

Contributors

- Pierre Baudot. (methods, theory, package and doc)
- Daniel Bennequin wikipedia (methods and theory)
- Mathieu Bernardi (convolutional patchs and GPU deployement - Machine Learning internship)
- Etienne Combrisson (application to CNS MRI, package and organization, documentation. . .)
- Jean-Marc Goaillard (genetic expression data application)
- Monica Tapia (genetic expression data application)