entropy

MDPI

Article

Estimating Differential Entropy using Recursive Copula Splitting

Gil Ariel * and Yoram Louzoun

Department of Mathematics, Bar Ilan University, Ramat Gan 5290002, Israel; louzouy@math.biu.ac.il
* Correspondence: arielg@math.biu.ac.il

Received: 22 January 2020; Accepted: 17 February 2020; Published: 19 February 2020

Abstract: A method for estimating the Shannon differential entropy of multidimensional random variables using independent samples is described. The method is based on decomposing the distribution into a product of marginal distributions and joint dependency, also known as the copula. The entropy of marginals is estimated using one-dimensional methods. The entropy of the copula, which always has a compact support, is estimated recursively by splitting the data along statistically dependent dimensions. The method can be applied both for distributions with compact and non-compact supports, which is imperative when the support is not known or of a mixed type (in different dimensions). At high dimensions (larger than 20), numerical examples demonstrate that our method is not only more accurate, but also significantly more efficient than existing approaches.

Keywords: entropy estimation; multivariate continuous distributions; copulas

1. Introduction

Differential entropy (DE) has wide applications in a range of fields including signal processing, machine learning, and feature selection [1–3]. DE estimation is also related to dimension reduction through independent component analysis [4], a method for separating data into additive components. Such algorithms typically look for linear combinations of different independent signals. Since two variables are independent if and only if their mutual information vanishes, accurate and efficient entropy estimation algorithms are highly advantageous [5]. Another important application of DE estimation is quantifying order in out-of-equilibrium physical systems [6,7]. In such systems, existent efficient methods for entropy approximation using thermodynamic integration fail and more fundamental approaches for estimating DE using independent samples are required.

The DE of a continuous multi-dimensional distribution with density $p(x) : \mathbb{R}^D \to \mathbb{R}$ is defined as,

$$H = -\int_{\mathbb{R}^n} p(x) \ln p(x) dx. \tag{1}$$

Despite a large number of suggested algorithms [8,9], the problem of estimating the DE from independent samples of distributions remains a challenge in high dimensions. Broadly speaking, algorithms can be classified as one of two approaches: Binning and sample-spacing methods, or their multidimensional analogues, namely partitioning and nearest-neighbor (NN) methods. In 1D, the most straight-forward method is to partition the support of the distribution into bins and either calculate the entropy of the histogram or use it for plug-in estimates [8,10,11]. This amounts to approximating $p(x)$ as a piece-wise constant function (i.e., assuming that the distribution is uniform in each subset in the partition). This works well if the support of the underlying distribution is bounded and given. If the support is not known or is unbounded, it can be estimated as well, for example using the minimal and maximal observations. In such cases, sample-spacing methods [8] that use the spacings between adjacent samples are advantageous. Overall, the literature provides a good arsenal

of tools for estimating 1D entropy including rigorous bounds on convergence rates (given some further assumptions of p). See [8,9] for reviews.

Estimating entropy in higher dimensions is significantly more challenging [9,12]. Binning methods become impractical as having M bins in each dimension implies M^D bins overall. Beyond the computational costs, most such bins will often have 1 or 0 samples, leading to the significant underestimating of the entropy. In order to overcome this difficulty, Stowell and Plumbley [13] suggested partitioning the data using a k-D partitioning tree-hierarchy (kDP). In each level of the tree, the data is divided into two parts with an equal number of samples. The splitting continues recursively across the different dimensions (see below for a discussion on the stopping criteria). The construction essentially partitions the support of p into bins that are multi-dimensional rectangles whose sides are aligned with the principal axes. The DE is then calculated assuming a uniform distribution in each rectangle. As shown below, this strategy works well at low dimensions (typically 2-3) and only if the support is known. The method is highly efficient, as constructing the partition tree has an $O(N \log N)$ efficiency. In particular, it has no explicit dependence on the dimension.

Spacing methods are generalized using the set of k nearest-neighbors to each sample (kNN) [11,14–17]. These are used to locally approximate the density, typically using kernels [10,18–22]. As shown below, kNN schemes preform well at moderately high dimensions (up to 10–15) for distributions with unbounded support. However, they fail completely when p has a compact support and become increasingly inefficient with the dimension. Broadly speaking, algorithms for approximating kNN in D-dimensions have an efficiency of $\epsilon^{-D} N \log N$, where ϵ is the required accuracy [23]. Other approaches for entropy estimation include variations and improvements of kNN (e.g., [4,19,22]), Voronoi-based partitions [24] (which are also prohibitively expensive at very high dimensions), Parzen windows [1], and ensemble estimators [25].

Here, we follow the approach of Stowell and Plumbley [13], partitioning space using trees. However, we add an important modification that significantly enhances the accuracy of the method. The main idea is to decompose the density $p(x)$ into a product of marginal (1D) densities and a copula. The copula is computed over the compact support of the one dimensional cumulative distributions. As such, the multidimensional DE estimates become the combination of one dimensional estimates, and a multi-dimensional estimate on a compact support, even if the support of the original distribution was not compact. We term the proposed method as copula decomposition entropy estimate (CADEE).

Following Sklar's theorem [26,27], any continuous multi-dimensional density $p(x)$ can be written uniquely as:

$$p(x) = p_1(x_1) \dots p_D(x_D) c(F_1(x_1), \dots, F_D(x_D)), \tag{2}$$

where, $x = (x_1, \dots, x_D)$, $p_k(\cdot)$ denotes the marginal density of the k'th dimension with the cumulative distribution function (CDF) $F_k(t) = \int_{-\infty}^{t} p_k(x) dx$, and $c(u_1, \dots, u_D)$ is the density of the copula, i.e., a probability density on the hyper-square $[0,1]^D$ whose marginals are all uniform on $[0,1]$,

$$\left[\Pi_{j=1 \dots D,\, j \neq k} \int du_j \right] c(u_1, \dots, u_D) = 1, \tag{3}$$

for all k. Substituting Equation (2) into Equation (1) yields,

$$H = \sum_{k=1}^{D} H_k + H_c, \tag{4}$$

where H_k is the entropy of the k'th marginal, to be computed using appropriate 1D estimators, and H_c is the entropy of the copula. Using Sklar's theorem has been previously suggested as a method for calculating the mutual information between variables, which is identical to the copula entropy H_c [5,28–30]. The new approach here is in showing that H_c can be efficiently estimated recursively, similar to the kDP approach.

Splitting the overall estimation into the marginal and copula contributions has several major advantages. First, the support of the copula is compact, which is exactly the premise for which partitioning methods are most adequate. Second, since the entropy of the copula is non-positive, adding up the marginal entropies across tree-levels provides an improving approximation (from above) of the entropy. Finally, the decomposition brings-forth a natural criterion for terminating the tree-partitioning and for dimension reduction using pairwise independence.

The following sections are organized as follows. Section 2 describes the outline of the CADEE algorithm. In order to demonstrate its wide applicability, several examples in which the DE can be calculated analytically are presented. In addition, our results are compared to previously suggested methods. Section 4 discusses implementation issues and the algorithm's computational cost. We conclude in section 5.

2. CADEE Method

The main idea proposed here is to write the entropy H as a sum of D 1D marginal entropies, and the entropy of the copula. Analytically, the copula is obtained by a change of variables,

$$c(u_1, \ldots, u_D) = p(F_1(x_1), \ldots, F_D(x_D)). \tag{5}$$

Let $x^i = (x_1^i, \ldots, x_D^i) \in \mathbb{R}^D, i = 1 \ldots N$ denote N independent samples from a real D-dimensional random variable (RV) with density $p(x)$. We would like to use the samples x^i in order to obtain samples from the copula density $c(u_1, \ldots, c_D)$. From Equation (5), this can be obtained by finding the rank (in increasing order) of samples along each dimension. In the following, this operation will be referred to as a rank transformation. This is the empirical analogue of the integral transform where one plugs the sample into the CDF. More formally, for each $k = 1 \ldots D$, let σ_k denote a permutation of $\{1 \ldots N\}$ that arranges $x_k^1, \ldots x_k^N$ in increasing order, i.e., $x_k^{\sigma_k^i} \leq x_k^{\sigma_k^j}$ for $i \leq j$. Then, taking

$$u_k^i = \frac{1}{N}(\sigma_k^i - 1/2), \tag{6}$$

yields N samples $u^i = (u_1^i, \ldots, u_D^i) \in [0,1]^D, i = 1 \ldots N$ from the distribution $c(u_1, \ldots, u_D)$. Note that the samples are not independent. In other words, the rank is the emperical CDF, shifted by $1/2N$. In particular, they correspond to N distinct points on a uniform grid, $u^i \in \{1/2N, 3/2N, 1 - 1/2N\}^D$.

1D entropies are estimated using either uniform binning or sample-spacing methods, depending on whether the support of the marginal is known to be compact (bins) or unbounded/unknown (spacing). The main challenge lies in evaluating the DE of high-dimensional copulas [5,31]. In order to overcome this difficulty, we compute it recursively, following the kDP approach. Let $k \in \{1, \ldots, D\}$ be spatial dimensions, to be chosen using any given order. The copula samples u^i are split into two equal parts (note that the median in each dimension is $1/2$). Denote the two halves as $v_j^i = \{u_j^i | u_k^i \leq 1/2\}$ and $w_j^i = \{u_j^i | u_k^i > 1/2\}$. Scaling the halves as $2v_j^i$ and $2w_j^i - 1$ produces two sample sets for two new copulas, each with $N/2$ points. A simple calculation shows that:

$$H_c = \frac{1}{2}(H_{2v} + H_{2w-1}), \tag{7}$$

where H_{2v} is the entropy estimate obtained using the set of points $2v_j^i$ and H_{2w-1} is the entropy estimate obtained using the set of points $2w_j^i - 1$. The marginals of each half may no longer be uniformly distributed in $[0,1]$, which suggests continuing recursively, i.e., the entropy of each half is a decomposed using Sklar's theorem, etc. See Figure 1 for a schematic sketch of the method.

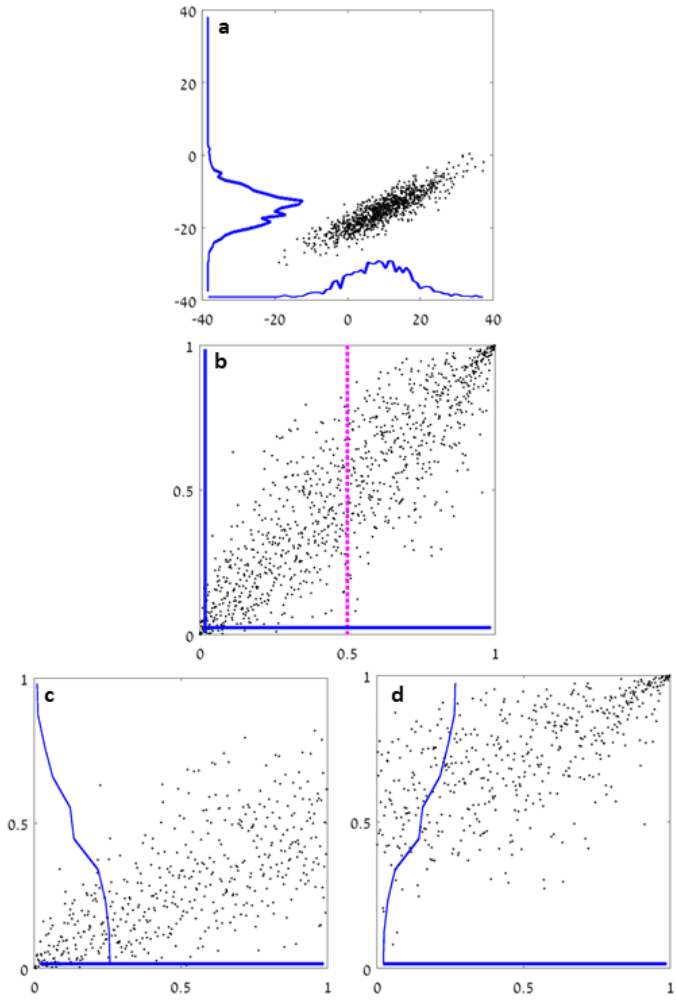

Figure 1. A schematic sketch of the proposed method. (**a**) A sample of 1000 points from a 2D Gaussian distribution. The blue lines depict the empirical density (obtained using uniform bins). (**b**) Following the rank transform (numbering the sorted data in each dimension), the same data provides samples for the copula in $[0,1]^2$. Splitting the data according to the median in one of the axes (always at 0.5) yields (**c**) (left half) and (**d**) (right half). The blue lines depict the empirical density in each half. They continue recursively.

A key question is finding a stopping condition for the recursion. In [13], Stowell and Plumbley apply a statistical test for uniformity of x_k, the dimension used for splitting. This condition is meaningless for our method as copulas have uniform marginals by construction. In fact, this suggests that one reason for the relatively poor *k*DP estimates at high D is the rather simplistic stopping criterion, requiring that only one of the marginals is statistically similar to a uniform RV.

In principle, we would like to stop the recursion once the copula cannot be statistically distinguished from the uniform distribution on $[0,1]^D$. However, reliable statistical tests for uniformity at high D are essentially equivalent to evaluating the copula entropy [5,18,31]. As a result, we relax the stopping condition to only test for pairwise dependence. The precise test for that will be further discussed. Calculating pairwise dependencies also allows a dimension reduction approach: If the matrix of pairwise-dependent dimensions can be split into blocks, then each block can be treated independently.

In order to demonstrate the applicability of the method described above, we study the results of our algorithm for several distributions for which the DE in Equation (1) can be computed analytically. Figures 2 and 3 show numerical results for H and the running time as a function of the dimension

using an implementation in Matlab. Five different distributions are studied. Three have a compact support in $[0,1]^D$ (Figure 2):

- **C1**: A uniform distribution;
- **C2**: Dependent pairs. The dimensions are divided into pairs. The density in each pair is $p(x,y) = x + y$, supported on $[0,1]^2$. Different pairs are independent;
- **C3**: Independent boxes. Uniform density in a set consisting of D small hypercubes, $\cup_{k=1}^{D}[(k-1)/D, k/D]^D$.

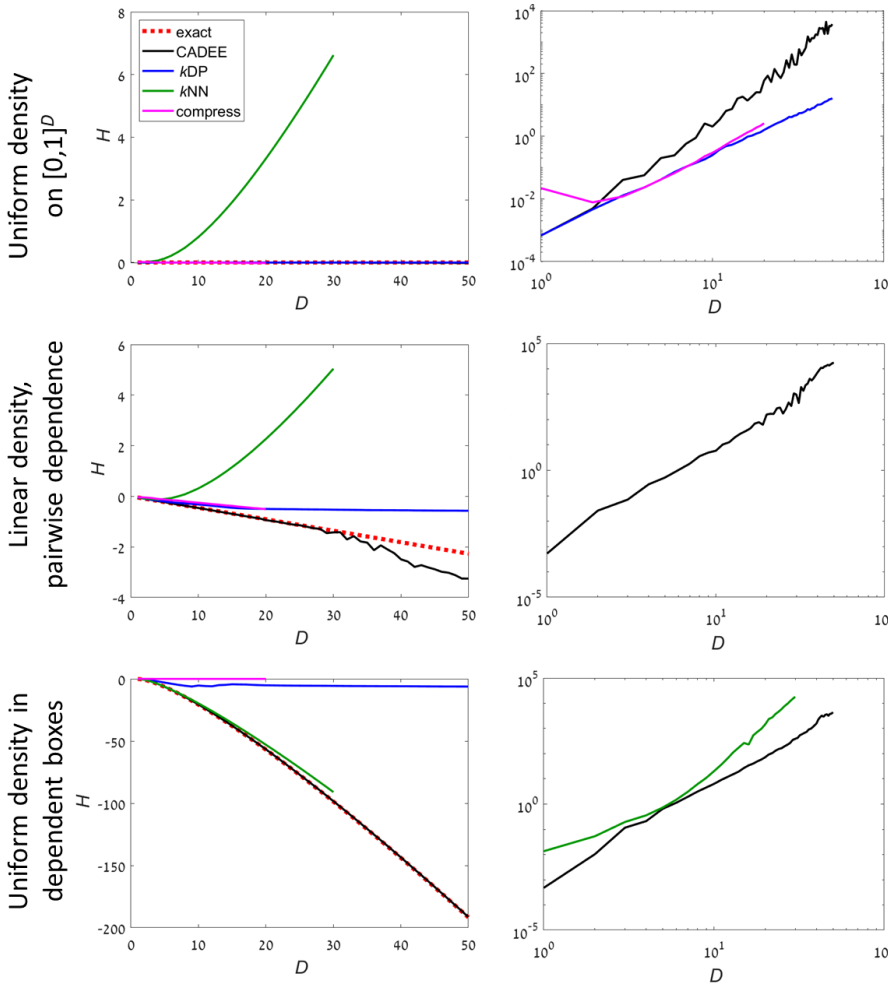

Figure 2. Estimating the entropy for given analytically-computable examples (dashed red line) with compact distributions ($[0,1]^D$). Black: Using the recursive copula splitting method, blue: kDP, green: kNN, and magenta: Lossless compression (magenta). (**Left**): The estimated entropy as a function of dimension. (**Right**): Running times (on a log-log scale), showing only relevant methods. The number of samples is $N = 10,000D^2$. See also Tables 1 and 2 for detailed numerical results with $D = 10$ and 20.

Two examples have an unbounded support (Figure 3):

- **UB1**: Gaussian distribution. The covariance is chosen to be a randomly rotated diagonal matrix with eigenvalues k^{-2}, $k = 1 \ldots D$. Then, the samples are rotated to a random orthonormal basis in $\mathbb{R}^D$. The support of the distribution is $\mathbb{R}^D$;
- **UB2**: Power-law distribution. Each dimension k is sampled independently from a density $x^{-2-2/k}$, $k = 1 \ldots D$ in $[1, \infty)$. Then, the samples are rotated to a random orthonormal basis in $\mathbb{R}^D$. The support of the distribution is a 2^{-D} fraction of $\mathbb{R}^D$ that is not aligned with the principal axes.

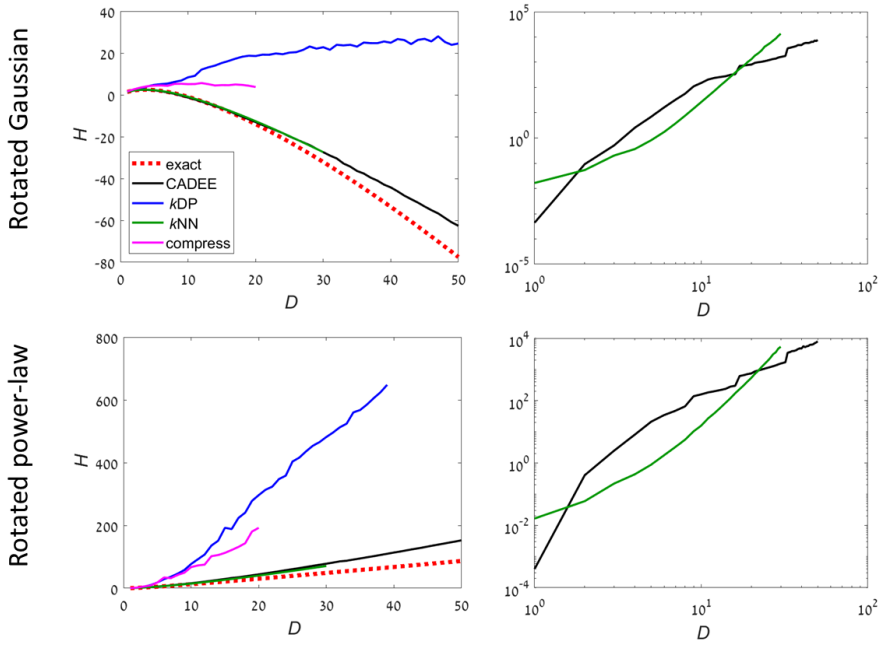

Figure 3. Estimating the entropy for given analytically-computable examples (dashed red line) with non-compact distributions. Black: Using the recursive copula splitting method, blue: *k*DP, green: *k*NN, and magenta: Lossless compression (magenta). (**Left**): The estimated entropy as a function of dimension. (**Right**): Running times (on a log-log scale), showing only relevant methods. The number of samples is $N = 10,000D^2$. The inaccuracy of our and the *k*NN method is primarily due to the relatively small number of samples. See also Tables 1 and 2 for detailed numerical results with $D = 10$ and 20.

Results with our method are compared to three algorithms:

1. The *k*DP algorithm [20]. We use the C implementation available in [32];
2. The *k*NN algorithm based on the Kozachenko–Leonenko estimator [14]. We use the C implementation available in [33];
3. A lossless compression approach [6,7]. Following [6], samples are binned into 256 equal bins in each dimension, and the data is converted into a $N \times D$ matrix of 8-bit unsigned integers. The matrix is compressed using the Lempel–Ziv–Welch (LZW) algorithm (implemented in Matlab's imwrite function to a gif file). In order to estimate the entropy, the file size is interpolated linearly between a constant matrix (minimal entropy) and a random matrix with independent uniformly distributed values (maximal entropy), both of the same size.

Theoretically, in order to get rigorous convergence of estimators, the number of samples should grow exponentially with the dimension [8]. Since this requirement is impractical at very high dimensions, we considered an under-sampled case and only used $N = 10,000D^2$ samples. Each method was tested at increasing dimensions until a running time of about 3 hours was reached (per run, on a standard PC) or the implementation ran out of memory. In such cases, no results are reported for this and following dimensions. See also Tables 1 and 2 for numerical results for $D = 10$ and 20.

Table 1. Estimating the entropy for given analytically-computable examples at $D = 10$. The best method is highlighted in bold.

Example	Exact	CADEE	*k*DP	*k*NN	Compression
C1—uniform	0	-1.5×10^{-3}	$\mathbf{-7.16 \times 10^{-4}}$	0.81	-4.3×10^{-2}
C2—pairs	−0.45	**−0.46**	−0.32	0.30	−0.25
C3—boxes	−20.7	**−20.6**	−5.3	−19.7	1.3×10^{-4}
UB1—Gauss	−0.915	−1.3	9.1	**−0.9**	5.1
UB2—power-law	12.6	15.7	92.3	**14.7**	67.2

Table 2. Estimating the entropy for given analytically-computable examples at $D = 20$. The best method is highlighted in bold.

Example	Exact	CADEE	*k*DP	*k*NN	Compression
C1—uniform	0	-2.9×10^{-3}	$\mathbf{-3.4 \times 10^{-4}}$	3.3	-1.1×10^{-2}
C2—pairs	−0.91	**−0.98**	−0.50	2.3	−0.43
C3—boxes	−56.9	−60.6	−5.15	**−52.9**	5.8×10^{-3}
UB1—Gauss	−14.0	**−14.4**	18.6	−12.6	5.0
UB2—power-law	30.2	47.2	296.6	**40.3**	131.6

Note that, in principle, it may be advantageous to apply a Principle Component Analysis (PCA) or Singular value Decomposition (SVD) of the sample convariance to decouple dependent directions. Such methods will be particular advantageous for the unbounded problems. We do not apply such conventional pre-processing methods here in order to make it more difficult for the CADEE method. If SVD converges the distribution into a product of independent 1D variables, the copula is close to 1 and the method will be highly exact after a single iteration.

For compact distributions, it is well known than *k*NN methods may fail completely. This can be seen even for the most simple examples such as uniform distributions (example C1). However, *k*NN worked well in example C3 because the density occupied a small fraction of the volume, which is optimal for *k*NN. *k*DP and compression methods are precise for uniform distribution, which is a reference case for these methods. For examples C2 and C3, both were highly inaccurate at $D > 5$. In comparison, CADEE showed very good accuracy up to D = 30–50, depending on the example.

For unbounded distributions, *k*DP and compression methods did not provide meaningful results for $D > 3$. Both CADEE and *k*NN provided good estimates up to $D = 20$ (*k*NN was slightly better), but diverged slowly at higher dimensions (CADEE was better). Numerical tests suggest this was primarily due to the relatively small number of samples, which severely under-sampled the distributions at high D. Comparing running times, the recursive copula splitting method was significantly more efficient at high dimensions. Simulations suggest a polynomial running time (see Section 4 for details), while *k*NN was exponential in D, becoming prohibitively inefficient at $D > 30$.

3. Convergence Analysis

In this section, we study the convergence properties of CADEE, i.e., the estimation error as N increases with fixed D. We proceeded along three routes. First, we considered an example in which the first several copula splittings could be preformed analytically. The example demonstrates how, ignoring statistical errors, recursive splitting of the copula and adding up the marginal entropies at the different recursion levels gets close to the exact entropy. Next, we provided a general analytical bound on the error of the model. Although the bound is not tight, it establishes that, in principle, the method provides a valid approximation of the entropy. Finally, we study the convergence of the method numerically for several low dimensional examples, providing empirical evidence that the rate of convergence of the method (the average absolute value of the error) is $O(N^{-\alpha})$ for some $0 < \alpha < 0.5$.

3.1. Analytical Example

In order to demonstrate the main idea why splitting the copula iteratively improved the entropy estimate, we worked-out a simple example in which the splittings could be performed analytically. For the purpose of this example, sampling errors in the estimate of the 1D entropy were neglected.

Consider the dependent pairs example (C2) with $D = 2$. The two dimensional density of the sampled random variable is given by:

$$p(x,y) = \begin{cases} x+y & 0 \le x,y \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The exact entropy is $H = -\int_0^1 \int_0^1 p \ln p dx dy = 5/6 - 4/3 \times \ln 2 \simeq -0.09086$. In order to obtain the copula, we first write the marginal densities and CDFs,

$$\begin{aligned} p_X(x) &= x + \frac{1}{2}, \quad F_X(x) = \frac{1}{2}x^2 + \frac{1}{2}x \\ p_Y(y) &= y + \frac{1}{2}, \quad F_Y(y) = \frac{1}{2}y^2 + \frac{1}{2}y \end{aligned}. \tag{9}$$

Using Sklar's theorem,

$$p(x,y) = p_X(x)p_Y(y)c(F_X(x), F_Y(y)). \tag{10}$$

Since the CDFs are invertible (in $[0,1]$), it can be equivalently written as,

$$p(F_X^{-1}(s), F_Y^{-1}(t)) = p_X(F_X^{-1}(s))p_Y(F_X^{-1}(t))c(s,t). \tag{11}$$

We invert the CDF's in Equation (11), $F_Y^{-1}(t) = F_X^{-1}(t) = (-1 + \sqrt{1+8t^2})/2$. Then substitute into Equation (11), hence,

$$c(s,t) = \frac{p(F_X^{-1}(s), F_Y^{-1}(t))}{p_X(F_X^{-1}(s))p_Y(F_X^{-1}(t))} = \frac{-1 + \sqrt{1/4+2s} + \sqrt{1/4+2t}}{\sqrt{1/4+2s}\sqrt{1/4+2t}}. \tag{12}$$

Indeed, one verifies that the marginals are uniform,

$$\int_0^1 c(x,y)dx = \int_0^1 c(x,y)dy = 1. \tag{13}$$

Continuing the CADEE algorithm, we computed the entropy of marginals, $H_X = H_Y = 1/2 - 9/8 \times \ln 3 + \ln 2 \simeq -0.04279$. This implies that the copula entropy is $H - H_X - H_Y \simeq -0.00528$ (5.8% of H). In order to approximate it, we split $c(x,y)$ into two halves, for example along the Y axis. Each density is shifted and stretched linearly to have support in $[0,1]^2$ again,

$$\begin{aligned} c_1(x,y) &= \frac{-1 + \sqrt{1/4+2x} + \sqrt{1/4+y}}{\sqrt{1/4+2x}\sqrt{1/4+y}} \\ c_2(x,y) &= \frac{-1 + \sqrt{1/4+2x} + \sqrt{5/4+y}}{\sqrt{1/4+2x}\sqrt{5/4+y}}. \end{aligned} \tag{14}$$

We continue recursively, computing the marginals for c_1 and c_2,

$$\begin{aligned} c_{1X}(x) &= \sqrt{5} - 1 + \frac{2-\sqrt{5}}{\sqrt{1/4+2x}}, \quad c_{1Y} = 1 \\ c_{2X}(x) &= 3 - \sqrt{5} + \frac{\sqrt{5}-2}{\sqrt{1/4+2x}}, \quad c_{2Y} = 1. \end{aligned} \tag{15}$$

The marginal entropies are $H_{1X} = -0.00284$, $H_{1Y} = 0$, $H_{1X} = -0.00267$, and $H_{2Y} = 0$. Overall, summing up the marginal entropies of the two iterations, we have $H_X + H_Y + 0.5(H_{1X} + H_{1Y} + H_{2X} + H_{2Y}) = -0.08834$ (error = 2.77%).

We similarly continue, calculating the copula of c_1 and c_2 and then the marginal distributions of their copulas. We found that the entropy after the third iteration is $H_X + H_Y + 0.5(H_{1X} + H_{1Y} + H_{2X} + H_{2Y} + 0.5(H_{11X} + H_{11Y} + H_{12X} + H_{12Y} + H_{21X} + H_{21Y} + H_{22X} + H_{22Y})) = 0.08993$ (error = 1.02%).

Indeed, we see that in the absence of statistical errors, the recursive splitting provides in improving upper bound for the entropy.

3.2. Analytical Bound

Here, we provide an analytical estimate of the bias and statistical error incurred by the algorithm. We derive a bound, which is not tight. Detailed analysis of the bias and error in some adequate norm is beyond the scope of the current paper.

The first part of the analysis estimated the worst-case accuracy by iteratively approximating the entropy using q repeated splittings of the copula. In the last iterations, the dimensions are assumed to be independent, i.e., the copula equals 1.

Consider the copula $c(u_1, \ldots, u_D)$, which is split, e.g., along $u_1 \in [0,1]$ into two halves corresponding to $u_1 \in [0, 1/2]$ and $u_1 \in [1/2, 1]$. Linearly scaling back into $[0,1]$, we obtain two densities:

$$\begin{aligned} c_1(s, u_2, \ldots, u_D) &= c(s/2, u_2, \ldots, u_D) \\ c_2(s, u_2, \ldots, u_D) &= c(1/2 + s/2, u_2, \ldots, u_D), \end{aligned} \tag{16}$$

where $(s, u_2, \ldots, u_D) \in [0,1]^D$. It is easily seen that $H_c = (H_{1c} + H_{2c})/2$, where H_{1c} and H_{1c} are the entropies of c_1 and c_2, respectively. We continue recursively, splitting the resulting copulas along some dimension. After q iterations, we obtain an expression of the form,

$$H = \sum_{j=1}^{D} \left[H_j + \sum_{k=1}^{q} \frac{1}{2^k} \sum_{i_1, \ldots, ß_k \in \{1,2\}} H_{i_1, \ldots, i_k, j} \right] + \frac{1}{2^q} \sum_{i_1, \ldots, ß_k \in \{1,2\}} H_{i_1, \ldots, i_k, c}, \tag{17}$$

where $H_{i_1,\ldots,i_k,j}$ is the 1D entropy of the j'th marginal and $H_{i_1,\ldots,i_k,c}$ is the entropy of the copula, obtained after k splittings along the dimensions $i_1, \ldots, ß_k$. For simplicity, we assume that the dimensions are chosen sequentially and suppose that $q = D^r$, i.e., each dimension was split r times.

Let $\Delta = 2^{-r}$ and suppose that the copula $c(x)$ is constant on small hyper-rectangles with sides:

$$[F_1^{-1}(i_1\Delta), F_1^{-1}((i_1+1)\Delta)] \times \cdots \times [F_D^{-1}(i_D\Delta), F_D^{-1}((i_D+1)\Delta)], \tag{18}$$

where $i_k \in \{0, \ldots, r-1\}$. This implies that within these rectangles all dimensions are independent. Then, $H_{i_1,\ldots,i_D,c} = 0$ and the last sum in Equation (17) vanishes.

Next, we approximate $c(x)$ in each small rectangle using Taylor. Without loss of generality, we focus on the case $i_1 = \cdots = i_D = 0$. To first order, $c(x) = (A + B_1 x_1 + \ldots B_D x_D)$, with $A, B_1, \ldots, B_D$ are $O(1)$. Scaling to $[0,1]^D$, $c_\Delta(x) = Z^{-1}(A + B_1 F_1^{-1}(\Delta)x_1 + \ldots B_D F_D^{-1}(\Delta)x_D)$, where Z is a normalization constant. Assuming that F_k are continuously differentiable and strictly increasing, F_K^{-1} are also continuously differentiable and $F_k^{-1}(\Delta) = O(\Delta)$. Then, since the total mass in each rectangle is exactly Δ^D, we have that $A/Z = 1 + O(\Delta)$. Finally, the entropy of the normalized density $c_\Delta(x)$ can be estimated. Expanding the log to order 1 in Δ,

$$H[c_\Delta] = -\int_0^1 dx_1 \cdots \int_0^1 dx_D c_\Delta(x) \ln c_\Delta(x) = -D \ln(1/\Delta) + O(\Delta). \tag{19}$$

From this, one needs to subtract $D \ln \Delta$ to compensate for the scaling. Therefore, for any continuously differential, strictly positive (in its support) density, $H_{i_1,\ldots,i_k,c} = O(\Delta)$. We conclude that the entire last sum in Equation (17) sums to order Δ. The prefactor is typically proportional to D.

Next, we consider statistical errors. Using the Kolmogorov–Smirnov statistics, the distance between the empirical CDF and the exact one is of order $N^{-1/2}$. Suppose 1D entropy estimates use a method with accuracy (absolute error) of order $N^{-\alpha}$, $\alpha \leq 1/2$. Then, in the worst case, if all errors are additive, then each estimate in the k'th iterate has an error (in absolute value) of order $(N/2^k)^{-\alpha}$. Overall, we have,

$$\begin{aligned} \Delta H &= D \sum_{k=1}^{q} \frac{1}{2^k} \sum_{i_1,\ldots,i_k \in \{1,2\}} \left(\frac{N}{2^k}\right)^{-\alpha} = D \sum_{k=1}^{q} \left(\frac{N}{2^k}\right)^{-\alpha} \\ &= DN^{-\alpha} \sum_{k=1}^{q} (2^\alpha)^k \leq DN^{-\alpha} (\sum_{k=1}^{q} 2^k)^\alpha = D(2^{q+1}/N)^\alpha. \end{aligned} \tag{20}$$

For fixed q, the statistical error decreases like $N^{-\alpha}$. Typically, for an unbiased 1D estimator in which the variance of the estimator is of order $N^{-2\alpha}$, the variance of the overall estimation using CADEE is,

$$\mathrm{Var}[\Delta H] = D(2^{q+1}/N)^{2\alpha}. \tag{21}$$

However, the prefactor depends linearly on the dimension D and exponentially on the number of iterations q. Recall that the bias decreases exponentially with q/D. Hence, the two sources of errors should be balanced in order to obtain a convergent approximation.

3.3. Numerical Examples

In order to demonstrate the convergence of the method, we test the error of the estimate obtained using CADEE for small D examples. Figure 4 shows numerical results with four types of distributions (dependent pairs, independent boxes, Gaussian, and power-law) with $D = 2$ and $D = 5$ and 10^3–10^8 samples. As discussed above, larger dimensions require significantly more samples in order to guarantee that the entire support is sampled at appropriate frequencies. We see that for all examples, the method indeed converged. For non-bounded distributions, the rate decreased with dimension.

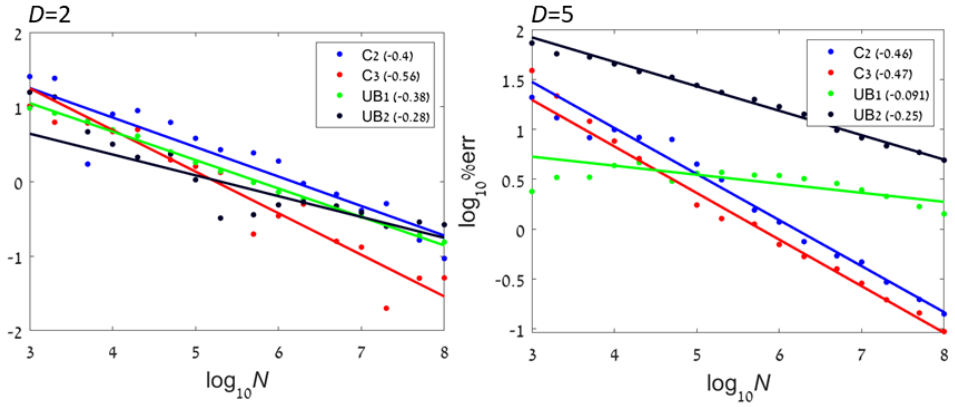

Figure 4. Convergence rates of CADEE: The average absolute value of the error as a function of N. (**Left**): $D = 2$. (**Right**): $D = 5$.

4. Implementation Details

The following is a pseudo-code implementation of the algorithm described above (Algorithm 1). Several aspects of the codes, such as choice of constants, stopping criterion, and estimation of pair-wise independence are rather heuristic approaches, which were found to improve the accuracy and efficiency

of our method. See Appendix A for details. Recall that for every i, $(x_1^i, \ldots, x_D^i) \in \mathbb{R}^D$ is an independent sample.

Algorithm 1 Recursive entropy estimator

function COPULAH($\{x_k^i\}$, D, N, $level = 0$)
  $H \leftarrow 0$
  for k = 1 to D **do**
    $u_k \leftarrow \text{rank}(x_k)/\text{N}$ ▷ Calculate rank (by sorting)
    $H \leftarrow H+$ H1D($\{u_k^i\}$, N, $level$) ▷ entropy of marginal k
  end for

  if $D = 1$ or $N <=$ min #samples **then**
    return H
  end if

  ▷ A is the matrix of pairwise independence
  $A_{ij} =$ true if x^i and x^j are statistically independent
  $n_{\text{blocks}} \leftarrow$ # of blocks in A.
  if $n_{\text{blocks}} > 1$ **then** ▷ Split dimensions
    for $j = 1$ to n_{blocks} **do**
      $v \leftarrow$ elements in block j
      $H \leftarrow H+$ copulaH($\{u_k^i\}_{k \in v}^{i=1 \ldots N}$,dim($v$),$N$,$level$)
    end for
    return H
  else ▷ No independent blocks
    $k \leftarrow$ choose a dim for splitting
    $L = \{i | u_k^i \leq 1/2\}$
    $\{v_j^i\} = \{2u_j^i | i \in L, j = 1 \ldots D\}$
    $H \leftarrow H+$ copulaH($\{v_j^i\}$,D,$N/2$,$level + 1$) /2

    $R = \{i | u_k^i > 1/2\}$
    $\{w_j^i\} = \{2u_j^i - 1 | i \in R, j = 1 \ldots D\}$
    $H \leftarrow H+$ copulaH($\{w_j^i\}$,D,$N/2$,$level + 1$) /2
  end if
end function

Several steps in the above algorithm should be addressed.

1. The rank of an array x is the order in which values appear. Since the support of all marginals in the copula is $[0,1]$, we take $\text{rank}(x) = \{1/2, 3/2, N - 1/2\}$. For example, $\text{rank}([-2, 0, -3]) = \{3/2, 5/2, 1/2\}$. This implies that the minimal and maximal samples are not mapped into $\{0, 1\}$, which would artificially change the support of the distribution. The rank transformation is easily done using sorting;
2. 1D entropy: One-dimensional entropy of compact distributions (whose support is $[0,1]$) is estimated using a histogram with uniformly spaced bins. The number of bins can be taken to depend of N, and order $N^{1/3}$ is typically used (we used $N^{1/3}$ or $N^0.4$ for spacing or bin-based methods, respectively. For additional considerations and methods for choosing the number of bins see [34]. At the first iteration, the distribution may not be compact, and the entropy is estimated using m_N-spacings (see [8], Equation (16));
3. Finding blocks in the adjacency matrix A: Let A be a matrix whose entries are 0 and 1, where $A_{kl} = 1$ implies that u^k and u^l are independent. By construction, A is symmetric. Let D denote the diagonal matrix whose diagonal elements are the sums of rows of A. Then, $L = A - D$ is the

Laplacian associated with the graph described by A. In particular, the sum of all rows of L is zero. We seek a rational basis for the kernel of a matrix L: Let $\ker(L)$ denote the kernel of a matrix L. By a rational basis we mean an orthogonal basis (for $\ker(L)$), in which all the coordinates are either 0 or 1 and the number of 1's is minimal. In each vector in the basis, components with 1's form a cluster (or block), which is pair-wise independent of all other marginals. In Matlab, this can be obtained using the command null(L,'r'). For example, consider the adjacency matrix:

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix},$$

whose graph Laplacian is:

$$D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad L = A - D = \begin{pmatrix} \text{-1} & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & \text{-1} \end{pmatrix},$$

A rational basis for the kernel of L (which is 2D) is:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\},$$

which corresponds to two blocks: Components 1+3 and component 2.

Pairwise independence is determined as follows:

1. Calculate the Spearman correlation matrix of the samples $\{x_k\}$, denoted R. Note that this is the same as the Pearson correlation matrix of the ranked data $\{u_k\}$;
2. Assuming normality and independence (which does not hold), the distribution of elements in R is asymptotically given by the t-distribution with $N-2$ degrees of freedom. Denoting the CDF of the t-distribution with n degrees of freedom by $T_n(z)$, two marginals (k,l) are considered uncorrelated if $|R_{kl}| > T_{n-2}^{-1}(1-\alpha/2)$, where α is the acceptance threshold. We take the standard $\alpha = 0.05$. Note that because we do $D(D-1)/2$ tests, the probability of observing independent vectors by chance grows with D. This can be corrected by looking at the statistics of the maximal value for R (in absolute value), which tends to a Gumbel distribution [35]. This approach (using Gumbel) is not used because below we also consider independence between blocks;
3. Pairwise independence using mutual information: Two 1D RVs X and Y are independent if and only if their mutual information vanishes, $I(X,Y) = H(X,Y) - H(X) - H(Y) = 0$ [10]. In our case, the marginals are $U(0,1)$ and $H(X) = H(Y) = 0$, hence $I(X,Y) = H(X,Y)$. This suggests a statistical test for the hypothesis that X and Y are independent as follows. Suppose X and Y are independent. Draw N independent samples and plot the density of the 2D entropy $H(X,Y)$. For a given acceptance threshold α, find the cutoff value $H_{2,c}$ such that $P(H(X,Y) < H_{2,c}) = 1-\alpha$. Figure A1 shows the distribution for different values of N. With $\alpha = 0.05$, the cutoff can be approximated by $H_{2,c} = -0.75N^{0.62}$. Accordingly, any pair of marginals which were found to be statistically uncorrelated, are also tested for independence using they mutual information (see below);
4. 2D entropy: Two-dimensional entropy (which, in our case, is always compact with support $[0,1]^2$) is estimated using a 2D histogram with uniformly spaced bins in each dimension.

As a final note, we address the choice of which dimension should be used for splitting in the recursion step. We suggest splitting the dimension which shows the strongest correlations with other marginals. To this end, we square the elements in the correlation matrix R and sum the rows. We pick the column with the largest sum (or the first of them if several are equal).

Lastly, we consider the computational cost of the algorithm, which has four components whose efficiency requires consideration:

1. Sorting of 1D samples: In the first level, samples may be unbounded and sorting can cost $O(N \log N)$. However, for the next levels, the samples are approximately uniformly distributed in $[0, 1]$ and bucket sort works with an average cost of $O(N)$. This is multiplied by the number of levels, which is $O(\log N)$. As all D dimensions need to be sorted, the overall cost of sorting is $O(DN \log N)$;
2. Calculating 1D entropies. Since the data is already sorted, calculating the entropy using either binning or spacing has a cost $O(N)$ per dimension, per level. Overall $O(DN \log N)$;
3. Pairwise correlations: $D(D-1)/2$ pre-sorted pairs, each costs $O(N)$ per level. Overall $O(D^2 N \log N)$;
4. Pairwise entropy: The worst-case is that all pairs are uncorrelated but dependent, which implies that all pairwise mutual information need to be calculated at all levels. However, pre-sorting again reduces the cost of calculating histograms to $O(N)$ per level. With $O(\log N)$ levels, the cost is $O(D^2 N \log N)$.

Overall, the cost of the algorithm is $O(D^2 N \log N)$. The bottleneck is due to the stopping criterion for the recursion. A simpler test may reduce the cost by a factor D. However, in addition to the added accuracy, checking for pairwise independence allows, for some distributions, splitting the samples into several lower dimensional estimates which is both efficient and more accurate.

5. Summary

We presented a new algorithm for estimating the differential entropy of high-dimensional distributions using independent samples. The method applied the idea of decoupling the entropy to a sum of 1D contributions, corresponding to the entropy of marginals, and the entropy of the copula, describing the dependence between the variables. Marginal densities were estimated using known methods for scalar distributions. The entropy of the copula was estimated recursively, similar to the k-D partitioning tree method. Our numerical examples demonstrated the applicability of our method up to a dimension of 50, showing improved accuracy and efficiency compared to previously suggested schemes. The main disadvantage of the algorithm was the assumption that pair-wise independent components of the data were truly independent. This approximations may clearly fail for particularly chosen setups. Rigorous proofs of consistency and analysis of convergence rates were beyond the scope of the present manuscript.

Our tests demonstrated that compression-based methods did not provide accurate estimates of the entropy, at least for the synthetic examples tested. Nonetheless, it was surprising that some quantitative estimate of entropy could be obtained using such simple-to-implement method. Moreover, this approach could be easily applied to high-dimensional distributions. Under some ergodic or mixing properties, independent sampling could be easily replaced by larger ensembles. Thus, for dimension 100 or higher (e.g., a 50 particles system in 2D), all the direct estimation methods (kDP, kNN, and CADEE) were prohibitively expensive.

To conclude, our numerical experiments suggest that kNN methods were favorable for unbounded distributions up to about dimension 20. At higher dimensions, kNN may become inaccurate, in particular for distributions with compact support (e.g., examples C1 and C2 in Figure 2). In addition, we found that kNN methods become inefficient at dimensions higher than 30 (e.g., examples UB1 and UB2 in Figure 3). For distribution with compact support, or when the support is mixed or unknown, the proposed CADEE method was significantly more robust. Our simple numerical examples suggest that the CADEE method may provide reliable estimates at relatively high dimensions (up to 100), even under severe under-sampling and at a reasonable computational cost. Here, we focused on the presentation of the algorithm and demonstrated its advantages for relatively simple analytically tractable examples. Applications to more realistic problems, for example estimating the entropy of physical systems that were out of equilibrium will be presented in a future publication. We suggest

using the recursive copula splitting scheme for other applications requiring estimation of copulas and evaluation of mutual dependencies between RVs, for example, in financial applications and neural signal processing algorithms.

A Matlab code is available in Matlab's File Exchange.

Author Contributions: Formal analysis, G.A. and Y.L.; Investigation, G.A. and Y.L.; Software, G.A.; Writing–original draft, G.A. and Y.L.; Writing–review and editing, G.A. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: G.A. thanks for the partial support of the Israel Science Foundation Grant No. 373/16 and the Deutsche Forschungsgemeinschaft (the German Research Foundation DFG) Grant No. BA1222/7-1.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Additional Pseudo-Code Used for Numerical Examples

Multiple methods can be used for estimating the 1D entropy, we applied the following pseudo-code.

Algorithm A1 Estimation of 1D entropy

function H1D($\{u^i\}$, N, *level*)
  if $level = 0$ **then**
    ▷ M_n Spacing method
    $M_n \leftarrow \text{round}(N^{1/3})$
    $\Delta^i \leftarrow u^{M_n+i} - u^i$, $i = 1, \ldots, N - M_n$
    $H \leftarrow \left[\sum_{i=1}^{M_n} \ln \Delta^i + (N - M_n) \ln(N/M_n)\right] / N$
  else ▷ Uniform bins in $[0, 1]$
    $N_{\text{bins}} \leftarrow \min\{\text{max\#bins}, N^{0.4}, N/10\}$
    $edges \leftarrow [0, 1/N_{\text{bins}}, 2/N_{\text{bins}}, \ldots, 1]$
    ▷ Histogram with bins *edges*
    $counts \leftarrow$ Histogram ($\{u^i\}$,*edges*)
    $p \leftarrow counts \cdot N_{\text{bins}} / (\sum counts)$ ▷ Normalize
    $H \leftarrow -\left[\sum p \ln(p)\right] / N_{\text{bins}}$ ▷ where $0 \ln 0 = 0$
  end if
  return H
end function

We suggest the following pseudo-code for estimating independence of two 1D RVs (already the rank vectors), which was used in the numerical examples.

Algorithm A2 Check for pairwise independence

function AREINDEPENDENT(X,Y,N)
  $R =$corr(X,Y) ▷ Pearson correlation
  $I \leftarrow$(P-value$(R, N) < \alpha$) ▷ true if statistically uncorrelated
  if I=false **then**
    $H_2 = \text{H2D}(X, Y)$ ▷ Calculate 2D entropy
    if $H_2 N^{0.62} < -0.75$ **then**
      $I \leftarrow$ true
    end if
  end if
end function

Algorithm A3 Estimation of 2D entropy

function H2D(X,Y,N)
    ▷ calc 2D histogram using uniform bins
    $N_{\text{bins}} \leftarrow \min\{\max\#\text{bins}, N^{0.2}, N/10\}$
    $edges \leftarrow [0, 1/N_{\text{bins}}, 2/N_{\text{bins}}, \ldots, 1]$
    ▷ Histogram with bins *edges* per dim
    $counts \leftarrow$ 2D Histogram(X,Y,*edges*)
    ▷ Normalize
    $p \leftarrow counts \cdot N_{\text{bins}}^2 / (\sum counts)$
    $H \leftarrow -\left[\sum p \ln(p)\right] / N_{\text{bins}}^2$
end function

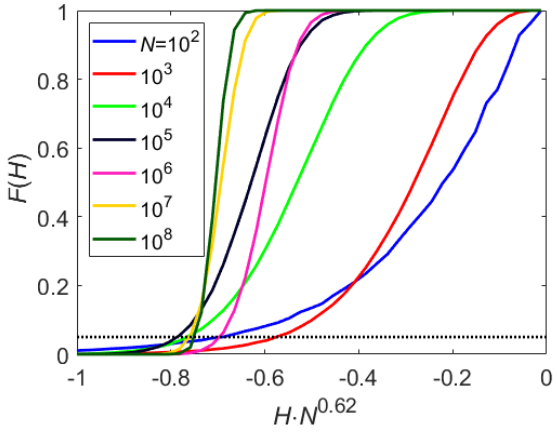

Figure A1. Numerical evaluation of the cumulative distribution function for the entropy of two scalar, independent, uniformly distributed random variables. After scaling with the sample size, we find that $P(HN^{-0.62} < -0.75)$ is approximately 0.05. Hence, it can be considered as a statistics for accepting the hypothesis that the random variables are independent.

References

1. Kwak, N.; Choi, C.-H. Input feature selection by mutual information based on parzen window. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1667–1671. [CrossRef]
2. Kerroum, M.A.; Hammouch, A.; Aboutajdine, D. Textural feature selection by joint mutual information based on gaussian mixture model for multispectral image classification. *Pattern Recognit. Lett.* **2010**, *31*, 1168–1174. [CrossRef]
3. Zhu, S.; Wang, D.; Yu, K.; Li, T.; Gong, Y. Feature selection for gene expression using model-based entropy. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2008**, *7*, 25–36.
4. Faivishevsky, L.; Goldberger, J. ICA based on a smooth estimation of the differential entropy. In Proceedings of the Advances in Neural Information Processing Systems 21 (NIPS 2008), Vancouver, BC, Canada, 8–10 December 2008; pp. 433–440.
5. Calsaverini, R.S.; Vicente, R. An information-theoretic approach to statistical dependence: Copula information. *Europhys. Lett.* **2009**, *88*, 68003. [CrossRef]
6. Avinery, R.; Kornreich, M.; Beck, R. Universal and accessible entropy estimation using a compression algorithm. *Phys. Rev. Lett.* **2019**, *123*, 178102. [CrossRef] [PubMed]
7. Martiniani, S.; Chaikin, P.M.; Levine, D. Quantifying hidden order out of equilibrium. *Phys. Rev. X* **2019**, *9*, 011031. [CrossRef]
8. Beirlant, J.; Dudewicz, E.J.; Györfi, L.; Van der Meulen, E.C. Nonparametric entropy estimation: An overview. *Int. J. Math. Stat. Sci.* **1997**, *6*, 17–39.
9. Paninski, L. Estimation of entropy and mutual information. *Neural Comput.* **2003**, *15*, 1191–1253. [CrossRef]
10. Granger, C.; Lin, J.L. Using the mutual information coefficient to identify lags in nonlinear models. *J. Time Ser. Anal.* **1994**, *15*, 371–384. [CrossRef]

11. Sricharan, K.; Raich, R.; Hero, A.O., III. Empirical estimation of entropy functionls with confidence. *arXiv* **2010**, arXiv:1012.4188.
12. Darbellay, G.A.; Vajda, I. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Trans. Inf. Theory* **1999**, *45*, 1315–1321. [CrossRef]
13. Stowell, D.; Plumbley, M.D. Fast multidimensional entropy estimation by $k-$d partitioning. *IEEE Signal Process. Lett.* **2009**, *16*, 537–540. [CrossRef]
14. Kozachenko, L.; Leonenko, N.N. Sample estimate of the entropy of a random vector. *Probl. Peredachi Informatsii* **1987**, *23*, 9–16.
15. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E* **2004**, *69*, 066138. [CrossRef] [PubMed]
16. Gao, W.; Oh, S.; Viswanath, P. Density functional estimators with k-nearest neighbor bandwidths. In Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT), Aachen, Germany, 25–30 June 2017; pp. 1351–1355.
17. Lord, W.M.; Sun, J.; Bollt, E.M. Geometric k-nearest neighbor estimation of entropy and mutual information. *Chaos* **2018**, *28*, 033114. [CrossRef]
18. Joe, H. Estimation of entropy and other functionals of a multivariate density. *Ann. Inst. Stat. Math.* **1989**, *41*, 683. [CrossRef]
19. Singh, H.; Misra, N.; Hnizdo, V.; Fedorowicz, A.; Demchuk, E. Nearest neighbor estimates of entropy. *Am. J. Math. Manag. Sci.* **2003**, *23*, 301–321. [CrossRef]
20. Shwartz, S.; Zibulevsky, M.; Schechner, Y.Y. Fast kernel entropy estimation and optimization. *Signal Process.* **2005**, *85*, 1045–1058. [CrossRef]
21. Ozertem, U.; Uysal, I.; Erdogmus, D. Continuously differentiable sample-spacing entropy estimates. *IEEE Trans. Neural Netw.* **2008**, *19*, 1978–1984. [CrossRef]
22. Gao, W.; Oh, S.; Viswanath, P. Breaking the bandwidth barrier: Geometrical adaptive entropy estimation. In Proceedings of the Advances in Neural Information Processing Systems 29 (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 2460–2468.
23. Indyk, P.; Kleinberg, R.; Mahabadi, S.; Yuan, Y. Simultaneous nearest neighbor search. *arXiv* **2016**, arXiv:1604.02188.
24. Miller, E.G. A new class of entropy estimators for multi-dimensional densities. In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, China, 6–10 April 2003.
25. Sricharan, K.; Wei, D.; Hero, A.O. Ensemble estimators for multivariate entropy estimation. *IEEE Trans. Inf. Theory* **2013**, *59*, 4374–4388. [CrossRef] [PubMed]
26. Jaworski, P.; Durante, F.; Hardle, W.K.; Rychlik, T. *Copula Theory and Its Applications*; Springer: New York, NY, USA, 2010.
27. Durante, F.; Sempi, C. Copula theory: An introduction. In *Copula Theory and Its Applications*; Springer: New York, NY, USA, 2010; pp. 3–33.
28. Giraudo, M.T.; Sacerdote, L.; Sirovich, R. Non–parametric estimation of mutual information through the entropy of the linkage. *Entropy* **2013**, *15*, 5154–5177. [CrossRef]
29. Hao, Z.; Singh, V.P. Integrating entropy and copula theories for hydrologic modeling and analysis. *Entropy* **2015**, *17*, 2253–2280. [CrossRef]
30. Xue, T. Transfer entropy estimation via copula. *Adv. Eng. Res.* **2017**, *138*, 887.
31. Embrechts, P.; Hofert, M. Statistical inference for copulas in high dimensions: A simulation study. *Astin Bull. J. IAA* **2013**, *43*, 81–95. [CrossRef]
32. Dan Stowell. $k-$d Partitioning Entropy Estimator: A Fast Estimator for the Entropy of Multidimensional Data Distributions. Available online: https://github.com/danstowell/kdpee (accessed on 16 February 2020).
33. Kalle Rutanen. TIM, A C++ Library for Efficient Estimation of Information-Theoretic Measures from Time-Series' in Arbitrary Dimensions. Available online: https://kaba.hilvi.org/homepage/main.htm (accessed on 16 February 2020).

34. Knuth, K.H. Optimal data-based binning for histograms. *arXiv* **2006**, arXiv:physics/0605197.
35. Han, F.; Chen, S.; Liu, H. Distribution-free tests of independence in high dimensions. *Biometrika* **2017**, *104*, 813–828. [CrossRef]