# *Experimental Results*

# Application of offset estimator of differential entropy and mutual information with multivariate data

Iván Marín-Franch, Martín Sanz-Sabater, and David H. Foster

**Supplementary Material**

# Appendix A. Proof of inherited asymptotic umbiasedness of the KLo estimator

## Let denote the expectation and the sample size, so that

Then, if denotes the absolute value of the determinant of the covariance matrix of ,

## as required.

# Appendix B. Description of software implementation

The KLo estimator was implemented in R, Python, and MATLAB computing environments. Each implementation used the environment’s corresponding library for the *K*-d tree nearest-neighbor search algorithm, but is otherwise self-contained. The same *K*-d tree parametrization was used throughout to ensure the compatibility of the numerical estimates in each environment.

The three packages are available in the GitHub repository at https://github.com/imarinfr/klo. There are two functions in each package. Their declarations are as follows, in R notation:

* entkl(x, type="klo", k=1): the KL estimator of differential entropy,
* mikl(x, y, type="klo", k=1): the KL estimator of mutual information,

where

* x, y are each *n*-by-*d* numeric matrices, in which the *n* rows correspond to observations and the *d* columns to variables (or coordinates) of the multivariate distributions;
* type is the type of estimator, "kl" for the Kozachenko-Leonenko estimator and "klo" (default value) for its offset version;
* k is the rank of the nearest neighbor to be searched for, where 1 is the nearest neighbor (default value), 2, the second nearest, and so on.

In the following, estimates of differential entropy and mutual information are given in bits.

# Appendix C. Examples

**Example 1. Worked example in R with color images**

Use of the packages at https://github.com/imarinfr/klo is straightforward, as this example with trivariant images illustrates for R (R Core Team, 2021). The supporting code can be found in the repository at example/example.r. To reduce computational time, the images were reduced in size by downsampling to 507×657×3 arrays.

Load the R packages R.Matlab and klo:

**library**(klo)

**library**(R.matlab)

Load each sample and reformat into three columns of LMS values lms1 and lms2:

R> lms1 <- **readMat**("../data/lms\_sete\_fontes\_1320.mat")$lms1

R> lms2 <- **readMat**("../data/lms\_sete\_fontes\_1321.mat")$lms2

R> nr <- **dim**(lms1)[1]

R> nc <- **dim**(lms1)[2]

R> nw <- **dim**(lms1)[3]

R> dim(lms1) <- c(nr \* nc, nw)

R> dim(lms2) <- c(nr \* nc, nw)

View the first few rows of lms1 and lms2. Values are less than unity because of the way LMS values are normalized.

R> **head**(lms1)

[,1] [,2] [,3]

[1,] 0.09121507 0.07145969 0.026087031

[2,] 0.07244906 0.05621339 0.018397820

[3,] 0.06147785 0.04765991 0.015102240

[4,] 0.05158587 0.04034708 0.012978409

[5,] 0.02914996 0.02340308 0.007394146

[6,] 0.01879230 0.01541703 0.004615444

R> **head**(lms2)

[,1] [,2] [,3]

[1,] 0.21220089 0.16694648 0.06104691

[2,] 0.17047076 0.13187180 0.04731434

[3,] 0.14673395 0.11324677 0.04046995

[4,] 0.12137365 0.09534710 0.03344379

[5,] 0.06750847 0.05440971 0.02147708

[6,] 0.04838272 0.03891643 0.01705063

Obtain the Gaussian component of the differential entropy for the first image:

R> **entg**(lms1)

-10.4435

These values are negative because the data values are less than unity. Obtain the KL estimates of differential entropy without and then with offset:

R> **entkl**(lms1, "kl")

-11.64424

R> **entkl**(lms1, "klo")

-11.61747

Obtain the Gaussian component of mutual information between the two images thus:

R> **mig**(lms1, lms2)

8.141937

The estimate is non-negative, as expected. Obtain the KL estimates of mutual information without and then with offset:

R> **mikl**(lms1, lms2, "kl")

8.92754

R> **mikl**(lms1, lms2, "klo")

9.91185

Notice that the KLo estimate is larger than the KL estimate, which in turn is larger than the Gaussian component.

# Example 2. Univariate example with stereophonic recording

This example illustrates the sample-size dependence of the KL and KLo estimators in comparison with one of the most widely used estimators (Singh & Póczos, 2016) of mutual information, the Kraskov-Stögbauer-Grassberger (KSG) estimator (Kraskov, Stögbauer, & Grassberger, 2004). The samples were derived from the left and right channels of a track from Kirsty MacColl’s album “Kite” (Deluxe Edition, 1989) of approximately 20 seconds duration, sample rate 32 kHz, 16-bit resolution The frequency distributions of the amplitude values were unimodal. The channels were highly correlated, and values were smoothed to lower the mutual information estimates slightly. Progressively larger samples, ranging from 210 to 219 points, were drawn randomly and identically from the two channels.

Figure S1 shows KLo, KL, and KSG estimates of the mutual information plotted against sample size. Each curve is an average over 100 repeated random samples. There is little difference between KLo and KL estimates, and both approached asymptotic values, but the KSG estimator performed poorly, consistent with the observation (Gao, Steeg, & Galstyan, 2015) that it tends to systematically underestimate mutual information with strongly dependent variables. The Gaussian component of the KLo estimator was very small, about 2.0 bits.

****

**Figure S1.** Estimates of mutual information between stereophonic channels. The plots show average mutual information estimates for each of the KLo, KL, and KSG estimators as a function of sample size. Standard deviations for the KLo and KL estimates ranged from about 0.2 with the smallest sample sizes to 0.007 with the largest sample sizes.

# References

Gao, S., Steeg, G. V., & Galstyan, A. (2015). *Efficient estimation of mutual information for strongly dependent variables*. Paper presented at the Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research. https://proceedings.mlr.press/v38/gao15.html

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E, 69*(6), 066138. https://doi.org/10.1103/PhysRevE.69.066138

R Core Team. (2021). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/

Singh, S., & Póczos, B. (2016). Finite-sample analysis of fixed-*k* nearest neighbor density functional estimators. *eprint arXiv:1606.01554*. https://doi.org/10.48550/arXiv.1606.01554