The Journal of Open Source Software

infotheory: A C++/Python package for multivariate information theoretic analysis

Madhavun Candadai[1, 2] **and Eduardo J. Izquierdo**[1, 2]

1 Cognitive Science Program, Indiana University, Bloomington, IN, U.S.A. **2** School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, U.S.A.

DOI: 10.21105/joss.01609

Software

- Review
- Repository
- Archive

Editor: Jack Poulson

Reviewers:

- @ajgates42
- @artemyk

Submitted: 04 July 2019
Published: 07 January 2020

Summary

This paper introduces `infotheory`: a package written in C++ and usable from Python and C++, for multivariate information theoretic analyses of discrete and continuous data. It is open-source (https://git.io/infot) and details on how to install it and use it are available on its website. This package allows the user to study the relationship between components of a complex system simply from the data recorded during its operation, using the tools of information theory. Specifically, this package enables the measurement of entropy, and mutual information, and also allows the user to perform partial information decomposition of mutual information into unique, redundant and synergistic information quantities.

Background

Information theory was first introduced by Claude Shannon in his seminal paper "A mathematical theory of communication" as a methodology to develop efficient coding and communication of data across noisy channels (Shannon, 1948). Its rise to popularity can be primarily attributed to its ability to be applied in any domain, ranging from Economics to Neuroscience. Information theory provides a general framework to quantify stochastic properties (uncertainty in the outcome of an experiment) and relationships (mutual information that one variable provides about another) between different variables in a system of interest. It provides tools to measure these quantities in a way that is invariant to the scale of the system and allows comparison across systems.

Statement of need

Until relatively recent times, information theory had been employed to study n-dimensional multivariate systems two variables at a time (bivariate). However, all natural systems are multivariate and a scientific inquiry into their operation requires understanding how these multiple variables interact. In a multivariate system, bivariate measures such as pairwise mutual information alone are insufficient to capture the polyadic interactions between the different variables (James & Crutchfield, 2017). Partial Information Decomposition (PID) is an extension of Shanon information measures that allows us to study the interaction between variables in a multivariate system by decomposing the total information that multiple source variables provide about a target variable into its constituent non-negative components (Williams & Beer, 2010). More specifically, in a trivariate case, the three variables can be separated into one target and two source variables. The total information that the two sources have about the target is given by the bivariate mutual information between the concatenated sources as one variable and the target. Using PID, the dependencies between the sources can be studied by decomposing this total information into the following non-negative components: information that each source uniquely provides about the target, information that they redundantly provide and the synergistic information that is only available when both sources are known. There have been multiple approaches proposed to perform said decomposition (Bertschinger, Rauh, Olbrich, Jost, & Ay, 2014; Griffith & Koch, 2014; R. G. James et al., 2018b; Williams & Beer, 2010). Here we focus on the approach proposed by (Williams & Beer, 2010) primarily because this package implements PID for two and three source decomposition, and as of now, this is the only approach that guarantees non-negative decomposition for the 4 variable case (1 target and 3 sources). Multivariate analysis allows us to ask more detailed questions such as, what is the amount of information that is uniquely provided about a target random variable by one source and not another? and what is the amount of information that is transferred from one random process X to another Y over and above Y's own information from its past? These questions enable us to understand the interactions between different components of a complex system, thereby leading us towards an understanding of its operation given just the observed data from the system.

Features

`infotheory` implements widely used measures such as entropy and mutual information (Cover & Thomas, 2012), as well as more recent measures that arise from multivariate extensions to information theory. As such, the tool has been designed to be easy to use and is ideal for pedagogical demonstrations of information theory as well as in research. Here, we highlight seven key aspects of its implementation that make our package a valuable addition to any information theoretic analyses tools set. First, the package is written in C++. One of the main challenges of multivariate analyses on a large, complex system is the amount of computations involved. Implementation in C++ makes the package efficient. Second, the package can be used from either C++ or Python. Python wrapping allows for ease of use, as well as compatibility with other powerful open-source libraries such as numpy. Third, the API allows adding the data only once to then perform various analyses across different subspaces of the dataset cheaply. Fourth, a custom sparse data structure is used to represent the random variables. This allows the package to work easily with large amounts of data. Fifth, to better estimate the data distribution in case of continuous variables, the package employs a kernel-based density estimation method called 'averaged shifted histograms' because of its beneficial trade-off between computational and statistical efficiency (Scott, 1985). Sixth, the package includes user-controllable specification of binning. This is essential for estimating distributions on hybrid systems with a mix of continuous and discrete variables. Finally, this package implements decomposition of information in 3 as well as 4 variable systems thus making it unique among similar existing packages.

One of the major challenges in utilizing information theoretic measures in experimental settings is the availability of sufficient data to infer the data distributions correctly (Paninski, 2003). To estimate data distribution from limited data, we have employed average shifted histograms for its beneficial trade-off between statistical and computational efficiency (Scott, 1985). This involves discretizing the data space into a number of bins and estimating frequentist probabilities based on the bins occupied by data samples. To reduce the impact of arbitrarily chosen bin boundaries the data distribution is estimated by averaging the bin occupancies across multiple shifted binnings of the data space. This binning based estimator has been shown to approximate a triangle kernel estimator (Scott, 1985). While the binning provides significant computational advantages, its approximation errors must be considered. Bias properties and guidelines for choosing the parameters for average shifted histograms are given in (Scott, 1979, 1985, p. @fernando2009selection, 2012). For a moderate sample size, 5 to 10 shifted histograms has been shown to be adequate (Scott, 1985). In general, average shifted histograms are best suited for noisy continuous data where the distribution of the data is unknown. For a more involved discussion on density estimation and its bias properties we point the reader to (Scott & Terrell, 1987) and (Wand & Jones, 1994).

Two existing packages that are most similar to ours are dit (R. G. James et al., 2018a) and IDTxL (Wollstadt et al., 2019). Unlike dit, our package can also help analyze continuous-valued data and unlike dit and IDTxL we have implemented PID analysis of 4 variables: 3 sources and 1 target. Other notable packages include: pyentropy (Ince, Bartolozzi, & Panzeri, n.d.), which was primarily designed for estimating entropies; and JDIT (Lizier, 2014), which was primarily designed for measuring transfer entropy. Neither pyentropy nor JDIT implement PID measures, although authors of JDIT have an unpublished GitHub repository that has a Java implementation of PID called JPID. In light of these existing packages and their functionalities, our package primarily focuses on measuring multivariate informational quantities on continuous data where the data distribution is not known a priori. However, it can still be used with discrete data using the same methods.

The functions implementing the above mentioned information theoretic measures have been designed to be flexibly used in alternative ways. For instance, the decomposed information components can be combined to measure transfer entropy (Schreiber, 2000). When dealing with time-series data, one can restructure the data such that the two sources are past values of two random variables, and the target is a future value of one of them. It has been shown that the sum of the unique information that a source provides about the target (future value) and the synergistic information from both sources is equal to the amount of information transferred from that source (Williams & Beer, 2011). Transfer entropy is used extensively in neuroscience to infer directed functional connections between nodes of a network (nodes can be neurons, brain regions or EEG electrodes) from recorded data (Wibral, Vicente, & Lindner, 2014). Another instance of extended use of this package is to measure changes in information in time. Again, with time-series data, if the user provides all data over all time-points, then they can ask the tool to calculate all the previously discussed measures as aggregate values over time. Alternatively, the user can provide data that are only from a specific time point, calculate the information theoretic measures for that time point, and then repeat the analyses over the entire time course. Such analysis reveals how information in the variables of the system change dynamically during the course of its operation (Beer & Williams, 2015; Izquierdo, Williams, & Beer, 2015). Both extensions are easily accessible by reusing the existing mutual information and PID functions in the package and providing different subsets of the data accordingly.

Conclusion

Altogether, `infotheory` provides an easy-to-use and flexible tool for performing information theoretic analyses on any multivariate dataset consisting of discrete or continuous data. Application areas are, in principle, as wide as that of information theory's - any domain that has a multivariate system and aims to study how the different components interact. We are particularly encouraged by the potential applications in neuroscience, at all scales ranging from individual neurons to brain regions to integrated brain-body-environment systems. In our group, we are currently using this package to understand the flow of information in simulated neural circuits capable of producing behavior. This tool allows us to easily analyze how different neurons of a circuit or regions in the brain are encoding information about the sensory stimulus it is receiving, the actions it is producing, or indeed about other neurons/regions within the system itself. We are using multivariate measures to analyze how different nodes in the circuit encode information uniquely, redundantly, and synergistically about a signal of interest. We are using the tool to study information dynamics of the neural circuit over time during behavior. We are also using it to infer directed functional connections between the nodes of the network. Besides its use in research, we are using this package for pedagogical purposes to introduce students to information theory. As such, we have provided a number of benchmarks and examples in the website. We also hope to continue to extend the package in the future by, for example, implementing additional approaches to multivariate information analyses, and providing GPU-support. Finally, in the spirit of free and open-source software development, we also welcome contributions from others.

Acknowledgements

The work in this paper was supported in part by NSF grant No. IIS-1524647. M.C. was funded by an assistantship from the Cognitive Science Program, Indiana University, Bloomington. The authors would like to thank Randall Beer for VectorMatrix, the C++ vector libraries used in this package.

References

Beer, R. D., & Williams, P. L. (2015). Information processing and dynamics in minimally cognitive agents. *Cognitive science*, *39*(1), 1–38. doi:10.1111/cogs.12142

Bertschinger, N., Rauh, J., Olbrich, E., Jost, J., & Ay, N. (2014). Quantifying unique information. *Entropy*, *16*(4), 2161–2183. doi:10.3390/e16042161

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons. doi:10.1002/047174882X

Fernando, T., Maier, H., & Dandy, G. (2009). Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. *Journal of Hydrology*, *367*(3-4), 165–176. doi:10.1016/j.jhydrol.2008.10.019v

Griffith, V., & Koch, C. (2014). Quantifying synergistic mutual information. In *Guided self-organization: Inception* (pp. 159–190). Springer. doi:10.1007/978-3-642-53734-9_6

Ince, R., Bartolozzi, C., & Panzeri, S. (n.d.). An information-theoretic library for the analysis of neural codes. doi:10.2417/1200906.1663

Izquierdo, E. J., Williams, P. L., & Beer, R. D. (2015). Information flow through a model of the c. Elegans klinotaxis circuit. *PloS one*, *10*(10), e0140397. doi:10.1371/journal.pone.0140397

James, R., & Crutchfield, J. (2017). Multivariate dependence beyond shannon information. *Entropy*, *19*(10), 531. doi:10.3390/e19100531

James, R. G., Ellison, C. J., & Crutchfield, J. P. (2018a). Dit: A python package for discrete information theory. *Journal of Open Source Software*, *3*(25), 738. doi:10.21105/joss.00738

James, R. G., Emenheiser, J., & Crutchfield, J. P. (2018b). Unique information via dependency constraints. *Journal of Physics A: Mathematical and Theoretical*, *52*(1), 014002. doi:10.1088/1751-8121/aaed53

Lizier, J. T. (2014). JIDT: An information-theoretic toolkit for studying the dynamics of complex systems. *Frontiers in Robotics and AI*, *1*, 11. doi:10.3389/frobt.2014.00011

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural computation*, *15*(6), 1191–1253. doi:10.1162/089976603321780272

Schreiber, T. (2000). Measuring information transfer. *Physical review letters*, *85*(2), 461. doi:10.1103/PhysRevLett.85.461

Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika*, *66*(3), 605–610. doi:10.1093/biomet/66.3.605

Scott, D. W. (1985). Averaged shifted histograms: Effective nonparametric density estimators in several dimensions. *The Annals of Statistics*, 1024–1040. doi:10.1214/aos/1176349654

Scott, D. W. (2012). Multivariate density estimation and visualization. In *Handbook of computational statistics* (pp. 549–569). Springer. doi:10.1007/978-3-642-21551-3_19

Scott, D. W., & Terrell, G. R. (1987). Biased and unbiased cross-validation in density estimation. *Journal of the american Statistical association*, *82*(400), 1131–1146. doi:10.1080/01621459.1987.10478550

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, *27*(3), 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x

Wand, M. P., & Jones, M. C. (1994). *Kernel smoothing*. Chapman; Hall/CRC. doi:10.1201/b14876

Wibral, M., Vicente, R., & Lindner, M. (2014). Transfer entropy in neuroscience. In *Directed information measures in neuroscience* (pp. 3–36). Springer. doi:10.1007/978-3-642-54474-3_1

Williams, P. L., & Beer, R. D. (2010). Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*.

Williams, P. L., & Beer, R. D. (2011). Generalized measures of information transfer. *arXiv preprint arXiv:1102.1507*.

Wollstadt, P., Lizier, J. T., Vicente, R., Finn, C., Martınez-Zarzuela, M., Mediano, P., Novelli, L., et al. (2019). IDTxl: The information dynamics toolkit xl: A python package for the efficient analysis of multivariate information dynamics in networks. *Journal of Open Source Software*, *4*(34), 1081. doi:10.21105/joss.01081