A Users' Guide for the MATALB/Python Package of 'Minimax Estimation of Functionals of Discrete Distributions' by Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman, IEEE Transactions on Information Theory, Vol.61, Issue 5, pp 2835-2885, May 2015. Version 2.0

October 21, 2015

Abstract

It is the users' guide for version 2.0 of the MATLAB/Python package of paper 'Minimax Estimation of Functionals of Discrete Distributions' by Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman, published in IEEE Transactions on Information Theory, Vol.61, Issue 5, pp 2835-2885, in May 2015. It demonstrates how to use the entropy and mutual information estimators developed in the paper in practice.

1 What is entropy and mutual information?

The Shannon entropy of a discrete distribution P is introduced by Shannon [1],

$$H(P) \triangleq \sum_{i=1}^{S} -p_i \ln p_i, \tag{1}$$

which plays significant roles in information theory and various disciplines such as statistics, machine learning, physics, neuroscience, computer science, linguistics, etc. Here the distribution P has support size S, which is assumed to be unknown.

The mutual information quantifies the dependence between two random variables, which for discrete random variables can be defined as

$$I(X;Y) = I(P_{XY}) = H(P_X) + H(P_Y) - H(P_{XY}), \tag{2}$$

where P_X, P_Y, P_{XY} denote, respectively, the distributions of random variables X, Y, and the pair (X,Y).

In applications, the true distribution P is usually unknown, hence we cannot compute these information theoretic measures. Instead, we consider the model where we obtain n independent identically distributed samples following distribution P, and would like to estimate the corresponding functionals such as entropy and mutual information.

Recently, Jiao, Venkat, Han, and Weissman [2] derived the first explicit entropy and mutual information estimators that achieve the minimax rates in the widest range of (S,n) (support size and sample size) pairs, and that do not require the knowledge of the support size. This MATLAB/Python package provides an efficient implementation of the estimators in [2].

2 What is new in Version 2.0?

Version 2.0 of this package added the following features:

1. Python implementation of the JVHW entropy and mutual information estimators
2. Error detection mechanism to detect whether the input sample only contains integers or not
3. Map unique values of the samples to small integers to avoid numerical overflow

3 How to use the estimators?

In the MATLAB implementation, there are two main functions that users may use: est_entro_JVHW.m and est_MI_JVHW.m. The Python implementation contains the counterpart of these two functions. We explain the usage of the MATLAB functions below.

3.1 est_entro_JVHW.m

est = est_entro_JVHW(samp)

This function returns the JVHW scalar estimate of the entropy (in bits) of samp when samp is a vector, and returns a row vector consisting of the JVHW entropy estimate of each column of samp when samp is a matrix. The input samp can only contain integers. The output est gives the JVHW entropy estimates (in bits) of the input vector or that of each column of the input matrix.

3.2 est_MI_JVHW.m

est = est_MI_JVHW(X,Y)

This function returns the JVHW estimate of the mutual information (in bits) of between each column of X and Y. Inputs X and Y must be of the same length, and must only contain integers. If X and Y are vectors, the function returns the JVHW estimate of the mutual information between these two random variables. If X and Y are matrices, the output est would be a row vector with each entry containing the JVHW estimate of the mutual information between the corresponding columns in X and Y.

The readers are welcomed to run the file test_entro.m to test the performance of the JVHW entropy and mutual information estimators. For comparison, we also provide the implementation of the maximum likelihood estimator of entropy (the empirical entropy), in function est_entro_MLE(samp) The way to use function est_entro_MLE(samp) is exactly the same as that of est_entro_JVHW(samp).

Note: The users may find that in test_entro.py we only evaluate the estimation performances over support size ranging from 10^2 to 10^5, rather than from 10^2 to 10^7 as in the MATLAB function test_entro.m. It is because sampling from a discrete distribution in Python is considerably slower than sampling in MATLAB. Our estimation algorithms have linear complexity and are thus efficient.

4 Acknowledgment

We are grateful to Igor Berman for his significant work in preparing the MATLAB version of this package. We are grateful to Yair Carmon for several valuable suggestions that speed up the code considerably. We are grateful to Wei Hu for implementing the Python version as well as increasing the robustness and efficiency of this package. We are grateful to Peng Liu

for rewriting the random sampling function as well as contributing various techniques that significantly improved the efficiency of the MATLAB code.

We have also implemented/downloaded another 10 entropy estimators (other than JVHW and MLE) in MATLAB that were used to compare with the JVHW estimator in [2, Section V]. The code is available under request.

References

[1] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[2] J. Jiao, K. Venkat, Y. Han, and T. Weissman, "Minimax estimation of functionals of discrete distributions," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2835–2885, 2015.