Uniform and Capacitated Mutual Information (UCMI) Estimator

Weihao Gao

June 28, 2016

1 Introduction

This document is a description of the Python package for Uniform and Capacitated Mutual Information (UCMI) estimator. You can download the code from . The paper describing UCMI can be found in [2]. The repository contains the following files:

- *ucmi.py*: Main code.
- *demo.py*: An example of usage.
- *readme.pdf*: Readme document.

2 Functions

1. **mi(x,y)**: Estimate the mutual information $I(X;Y)$ of $X \in \mathbb{R}^{d_x}$ and $Y \in \mathbb{R}^{d_y}$ from samples $\{x_i, y_i\}_{i=1}^N$ using KSG estimator [1].
   - **x**: A 2D list of dimension $N \times d_x$, where each row is one sample $x_i \in \mathbb{R}^{d_x}$.
   - **y**: A 2D list of dimension $N \times d_y$, where each row is one sample $y_i \in \mathbb{R}^{d_y}$.
   - Output: Scalar $\widehat{I}(X;Y)$.
2. **umi(x,y)**: Estimate the uniform mutual information $\mathrm{UMI}(X;Y)$ [2] of $X \in \mathbb{R}^{d_x}$ and $Y \in \mathbb{R}^{d_y}$ from samples $\{x_i, y_i\}_{i=1}^N$.
   - **x**: A 2D list of dimension $N \times d_x$, where each row is one sample $x_i \in \mathbb{R}^{d_x}$.
   - **y**: A 2D list of dimension $N \times d_y$, where each row is one sample $y_i \in \mathbb{R}^{d_y}$.
   - Output: Scalar $\widehat{\mathrm{UMI}}(X;Y)$.
3. **cmi(x,y)**: Estimate the capacitated mutual information $\mathrm{CMI}(X;Y)$ [2] of $X \in \mathbb{R}^{d_x}$ and $Y \in \mathbb{R}^{d_y}$ from samples $\{x_i, y_i\}_{i=1}^N$.
   - **x**: A 2D list of dimension $N \times d_x$, where each row is one sample $x_i \in \mathbb{R}^{d_x}$.
   - **y**: A 2D list of dimension $N \times d_y$, where each row is one sample $y_i \in \mathbb{R}^{d_y}$.
   - Output: Scalar $\widehat{\mathrm{CMI}}(X;Y)$.

3 Usage

Here we provide a simple sample of usage of the package. Here X and Y are independent standard Gaussian random variable, so $I(X;Y) = \mathrm{UMI}(X;Y) = \mathrm{CMI}(X;Y) = 0$.

```
>> import numpy.random as nr
>> import ucmi
>> x = nr.normal(0,1,[1000,1])
>> y = nr.normal(0,1,[1000,1])
>> print "I(X;Y) = ", ucmi.mi(x,y)
I(X;Y) = 0.000845730994389
>> print "UMI(X;Y) = ", ucmi.umi(x,y)
UMI(X;Y) = -0.0469967469206
>> print "CMI(X;Y) = ", ucmi.cmi(x,y)
CMI(X;Y) = -0.00383973235348
```

You can find the code in *demo.py*.

References

[1] Kraskov A, Stögbauer H, Grassberger P. Estimating mutual information[J]. Physical review E, 2004, 69(6): 066138.

[2] Gao, W., Kannan, S., Oh, S. and Viswanath, P. Conditional Dependence via Shannon Entropy: Axioms, Estimators and Applications, International Conference for Machine Learning, 2016.