Local Nearest Neighbor (LNN) Information Estimator

Weihao Gao

September 10, 2016

1 Introduction

This document is a description of the Python package for Local Nearest Neighbor (LNN) Information estimator. You can download the code from http://github.com/liverlover/lnn/. The paper describing LNN estimators can be found in [1]. The repository contains the following files:

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

2 Functions

2.1 Main Functions

Here are the proposed entropy and mutual information estimator in [1]:

1. **entropy(data)**: Estimate the differential entropy $H(X)$ of $X \in \mathbb{R}^{d_x}$ from samples $\{x_i\}_{i=1}^N$ using LNN entropy estimator [1, Section 3].
   - **data**: A 2D list of dimension $N \times d_x$, where each row is one sample $x_i \in \mathbb{R}^{d_x}$.
   - Output: Scalar $\widehat{H}(X)$.
2. **mi(data,split)**: 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 3LNN mutual information estimator [1, Section 5].
   - **data**: A 2D list of dimension $N \times (d_x + d_y)$, where each row is one pair of sample $(x_i, y_i) \in \mathbb{R}^{d_x+d_y}$.
   - **split**: Equals to d_x, telling which part of data represent X and which part represent Y.
   - Output: Scalar $\widehat{I}(X;Y)$.

2.2 Other Estimators for Comparison

Here we also provide other entropy and mutual information estimators which are used for comparison in the experiments in [1].

- *Entropy Estimators*:
  1. **KDE_entropy(data):** KDE entropy estimator [2].
  2. **KL_entropy(data):** Kozachenko-Leonenko entropy estimator [3].
  3. **LNN_1_entropy(data):** LNN entropy estimator with order parameter $p = 1$.
- *Mutual Information Estimators*:
  1. **_3KDE_mi(data,split):** Combination of 3 KDE entropy estimators.

2. **_3KL_mi(data,split):** Combinations of 3 Kozachenko-Leonenko entropy estimator.
3. **_KSG_mi(data,split):** KSG mutual information estimator [4].
4. **_3LNN_1_mi(data,split):** Combination of 3 LNN entropy estimators with $p = 1$.
5. **_3LNN_1_KSG_mi(data,split):** Combinations of 3 LNN entropy estimators with $p = 1$, using "KSG trick" (see [1, Section 5]).
6. **_3LNN_2_KSG_mi(data,split):** Combinations of 3 LNN entropy estimators with $p = 2$, using "KSG trick" (see [1, Section 5]).

3 Usage

Here we provide a simple sample of usage of the package. Here X and Y are joint standard Gaussian random variable with high correlation.

```
>> import numpy.random as nr
>> from math import log, pi, exp
>> import lnn
>> r = 0.9999
>> data = nr.multivariate_normal([0,0],[[1,r],[r,1]],100)
>> print "Ground Truth = ", log(2*pi*exp(1))+0.5*log(1-r**2)
Ground Truth = -1.42074452992
>> print "LNN: H(X) = ", lnn.entropy(data)
LNN: H(X) = -1.38841327294
```

You can find the full version of the code in *demo.py*.

References

[1] Gao, W., Oh, S. and Viswanath, P. Breaking the Bandwidth Barrier: Geometrical Adaptive Entropy Estimation, Conference on Neural Information Processing Systems (NIPS), 2016.

[2] Beirlant J, Dudewicz E J, Györfi L, et al. Nonparametric entropy estimation: An overview[J]. International Journal of Mathematical and Statistical Sciences, 1997, 6(1): 17-39.

[3] L. F. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. Problem Peredachi Informatsii, 23(2):916, 1987.

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