Information Theory and Coding: Computer Assignment

Behrad Moniri (95109564)
bemoniri@live.com

1 Estimating Mutual Information: Review

In this section, we will review methods of estimating mutual information from finite samples and we will implement the KDR algorithm to estimate the mutual information and test the correctness of the algorithm in a simulation.

1.1 The Naive Approach: Binning and Histograms

The most naive method of estimating mutual information is by estimating a histogram and using the histogram for mutual information estimation. This method is know to be significantly biased.

$$I(X,Y) \approx I_{\text{binned}}(X,Y) = \sum_{i,j} p(i,j) \log\Big(\frac{p(i,j)}{p_x(i)_y(j)}\Big)$$

Here, p_x, p_y and p are estimations of the probability distribution function using histograms. Besides being biased, this method also is very dependent on the bin size [1, 2].

1.2 Adaptive Binning

This method is an extension of the naive method mentioned above. This method is widely used the signal processing literature for blind source separation. In contrast to the method described above, here the main idea is choose the bin size adaptability. The idea is to put more emphasis on areas with more variation in input space.

Initially, we consider a one-cell partition involving all data pairs. In every iteration, cells are divided into equiprobable halves. The partitioning of each cell is accepted unless the ration $\frac{f_{XY}(x,y)}{f_X(x)f_Y(y)}$ takes approximately the same value in each of the four sub-cells as it does in the cell itself [3].

1.3 k Nearest Neighbor (kNN) Based Methods

Given i.i.d. samples $X_1, \dots, X_n$ generated from density p_X, the problem is to estimate p_X at a given point x_0. The k-NN of x_0 in $X_1, ..., X_n$ is $X_{i(k)}$, where $i(1), \dots, i(n)$ is such that

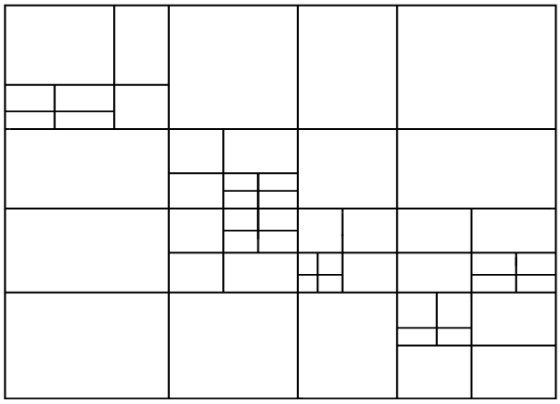

Figure 1: Example of Adaptive Bin Size

$||x_0 - X_{i(1)}|| \leq ||x_0 - X_{i(2)}|| \leq \cdots \leq ||x_0 - X_{i(n)}||$. Let ρ_k be the distance between x_0 and the k-th neighbor. The density p_X can be estimated as:

$$\hat{p}_X(x) = \frac{k(n)/n}{\lambda(B_{x,\rho_k})}$$

Where B_{x,ρ_k} is the ball centered at x with radius ρ.
Theorem 1.

$$\lim_{n\to\infty} |\hat{p}_X(x) - p_X(x)| \to 0, \quad \text{almost surely}$$

provided $\frac{k(n)}{\log n} \to \infty$ *and* $\frac{n}{k(n)} \to \infty$.

We will use the following notion of distance in the space $\mathbb{R}^{d_X+d_Y}$ i which $z = (x, y)$:

$$||z - z'|| = \max\{||x - x'||, ||y - y'||\}$$

Denote by $\epsilon(i)/2$ the distance from z ito its k-NN, and by $\epsilon(x(i)/2)$ and $\epsilon(y(i)/2)$ the distances between the same points projected into the X and Y sub-spaces. $n_x(i)$ is the number of points whose distance from x_i is strictly less than $\epsilon(i)/2$, andn $y(i)$ is similarly defined for y. The estimate for $I(X;Y)$ is

$$I(X;Y) \approx -\frac{1}{n}\sum_{i=1} n[\psi(n_x(i)+1) + \psi(n_y(i)+1)] + \psi(k) + \psi(n)$$

where ψ is the Digamma function [1].

1.4 Kernel Density Estimation (KDE)

Another method for estimating mutual information is kernel density estimation. The density of a random variable (vector) X can be estimated from i.i.d. samples, $X_1, X_2, \ldots, X_n$ with the following equations:

$$\hat{p}_X(z) = \frac{1}{nw_n^{d_X}} \sum_{i=1}^{n} K\Big(\frac{z - X}{w_n}\Big)$$

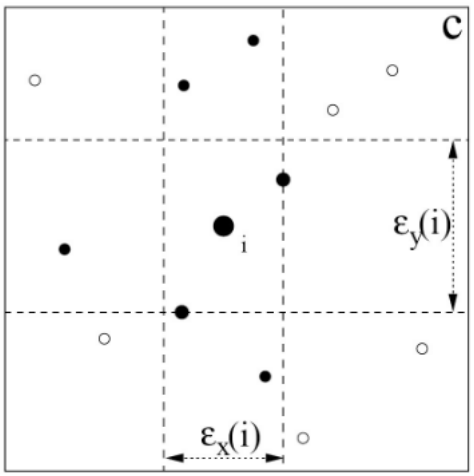

Figure 2: an example of kNN method with k = 1

where d_X is the dimension of X. w_n controls the width of the kernel and w_n is a sequence convergent to zero.

Given this estimate, we can estimate the mutual information:

$$\hat{I}(X;Y) = \frac{1}{n}\sum_{i=1}^{n} \log \frac{\hat{p}_{XY}(X_i, Y_i)}{\hat{p}_X(X_i)\hat{p}_Y(Y_i)}$$

This results are from [2].

1.5 Choosing the algorithm to implement

I have been working on BSS and Sparse Recovery as my BSc thesis and I am familiar with estimators of mutual information. To my knowledge (gained over the time I worked on my BSc thesis), given a large sample size, the kernel estimation method gives the most accurate results for estimating mutual information. For this reason, we will implement this algorithm. The adaptive binning method is also very efficient and the time complexity is very low, but very good implementations of this algorithm already exist. We will use this algorithm (the implementation of adaptive binning algorithm is not done me!) to compare its results with the results of the algorithm I have implemented.

2 Implementation of KDE and Simulations

By comparing the simulation results of the mentioned methods in the literature, we decided to implement the KDE algorithm. We will use the gaussian kernel for estimation.

The generative model of the data is:

$$\begin{cases} X = N_1 \\ Y = X + N_2 \end{cases}$$

where N_1 and N_2 are independent standard Gaussian random variables. In theory we know that the mutual information of two jointly Gaussian random variables is $-\frac{1}{2}\ln(1-\rho^2)$. The covariance matrix of our data is $\begin{bmatrix} 15 & 15 \\ 15 & 16 \end{bmatrix}$, thus $I(X;Y) = 0.3466$. We will use a sample size of 1024 for estimation. Besides using our own implementation of KDE, we also use a well known implementation of the adaptive algorithm. . From theory we know that:

$$I(X;Y) = \frac{1}{2}\log_2(1 + \frac{P}{N})$$

The results are presented in table 1. These results are from [2].

Theory	KDE	Adaptive
2.000	2.5525	2.4302

Table 1: Simulation Results

3 Estimating Channel Capacity and the Blahut-Arimoto Algorithm

In this section, we will implement the Blahut-Arimoto algorithm to estimate the channel capacity [4]. This algorithm is based on alternative optimization and update of the distribution and an auxiliary variable. **I have read and understood the reasoning behind BL algorithm, but I will not repeat the theorems and results here, all can be found in [4]. I can explain this algorithms to the teaching assistants when requested!**

The Blahut-Arimoto is presented below:

Init: Initialize $p(1)$ to the uniform distribution over χ, i.e. $p_i(1) = \frac{1}{|\chi|}$

Step 2: Find $\phi^{(t+1)}$ as follows:

$$\phi^{(t+1)} = \frac{p_i^{(t)} Q_{ij}}{\sum_k p_k^{(t)} Q_{kj}}$$

Step 3: Update $p^{(t+1)}$ as follows:

$$p_i^{(t+1)} = \frac{r_i^{(t+1)}}{\sum_{k\in\chi} r_k^{(t+1)}}$$

3.1 Experimental Results

3.1.1 Channel A

In this channel the transition matrix is

$$\begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$$

The output of the BL algorithm is: $\hat{C} = 0.3651$

3.1.2 Channel B

There must be a typo in the assignment. Rows of the transition matrix of this channel do not have unit sum.

3.1.3 Channel C

In this channel the transition matrix is

$$\begin{bmatrix} 0.7 & 0.3 & 0 \\ 0 & 0.3 & 0.7 \end{bmatrix}$$

The output of the BL algorithm is: $\hat{C} = 0.7000$. This is correct as the channel is a BEC with error probability 0.3, thus the capacity is $C = 0.7$.

References

[1] Qing Wang, Sanjeev R Kulkarni, Sergio Verdú, et al. Universal estimation of information measures for analog sources. *Foundations and Trends® in Communications and Information Theory*, 5(3):265–353, 2009.

[2] Christopher J Cellucci, Alfonso M Albano, and Paul E Rapp. Statistical validation of mutual information calculations: Comparison of alternative numerical algorithms. *Physical Review E*, 71(6):066208, 2005.

[3] Georges A Darbellay. Independent component analysis through direct estimation of the mutual information.

[4] Raymond W Yeung. The blahut–arimoto algorithms. In *Information Theory and Network Coding*, pages 211–228. Springer, 2008.