Information Bottleneck & Information Estimators

Moulik Choraria
Master's in Communication Systems
EPFL, Switzerland

Dr. Michael C. Gastpar
Laboratory for Information in Networked Systems
Full Professor at EPFL, Switzerland

***Abstract*—The information bottleneck method provides a structured framework for deriving minimally sufficient statistics for a probability distribution. In practice however, the representation needs to be learned from finite samples. Consequently, the optimal solution inherently depends on the chosen information estimator. This project aims to understand the relationship between the same.**

I. Introduction

A. The Information Bottleneck Method

The Information Bottleneck method, first introduced by Tishby et al. [1], provides a formulation to extract a relevant summary of a random variable X. Here, the relevant information in X is quantified by means of the mutual information between X and a relevance variable Y. Thus, the learned representation $\widetilde{X}$ is privy to the Markov chain $(Y \leftarrow X \leftarrow \widetilde{X})$. Formally, the formulation takes the form below:

$$L[p(\widetilde{X}|X)] = I(\widetilde{X};X) - \beta I(\widetilde{X};Y) \tag{1}$$

The loss function is to be minimized over the space of all conditional distributions $p(\widetilde{X}|X)$ and a solution using the iterative Blahut-Arimoto algorithm may be obtained, the uniqueness of which is not guaranteed.

B. Information Measure Estimators

There exist several methodologies in literature for the estimation of entropy and mutual information from finite samples, from histogram and kernel density estimates to nearest neighbour approaches [2]. The estimators achieve reasonable accuracy on different problems, with inherent simplifying assumptions which seem to hold reasonably well in a variety of different situations.

The aim of this study is to explore the problem of Information Bottleneck, through the lens of the information theoretic measure estimators and try and determine possible underlying structural relationships in the optimal solution pertaining to the particular estimator.

II. Information Estimation: Review

Shannon Entropy of a random variable X with $\mu(x)$ as the probability density function, is defined as follows:

$$H(X) = -\int \mu(x) log(\mu(x)) dx \tag{2}$$

Entropy is thus an expectation of $log(\mu(x))$ and with an unbiased estimator of $\mu(x)$ obtained from N samples, one may obtain an estimate of entropy by the law of large numbers as follows:

$$\widehat{H}(X) = -N^{-1} \sum_{i=1}^{N} log(\widehat{\mu(x_i)}) \tag{3}$$

The mutual information may then be obtained as the difference between the sum of marginal entropies and the joint entropy. To obtain an estimate for the density function, several approaches are possible.

Histogram Estimator: The simplest estimator, where the sample space is divided into a grid of appropriately sized bins, and each sample is placed into their corresponding bins. Dividing the number of samples in each bin by the total number of samples gives a discrete probability distribution, which gives a discrete estimate for the underlying probability distribution.

Kernel Density Estimator: To improve upon the previous class of estimators and obtain a smooth density function, the estimate is obtained by choosing an appropriate kernel, for instance Gaussian, and its parameters. The kernels are centred around each sample and the aggregate sum function over all samples gives the estimate for the density. Empirical rules exist for determining the parameters of the kernels. The the entropy estimate is obtained by either integrating over the entire function, or re-substituting values from the obtained density into the estimator in equation (3).
One of the major drawbacks of the first two classes of estimators is the explicit requirement of choosing the bin sizes/bandwidths appropriately, which makes them susceptible to the curse of dimensionality problem as well as scaling of the distribution. The nearest neighbour class of estimators, based on the estimator first proposed by Kozachenko et al. [3] rely on an implicit estimate of the density function. They also do away with the need to choose an appropriate bin size, thus alleviating the curse of the dimensionality problem to some extent.

K-NN Estimator: For the purpose of this study, the mutual information estimator proposed by Kraskov et al. [4] based on nearest neighbour estimates is chosen. The idea is to estimate the underlying distribution using the the k-nearest neighbour statistics. First, a new variable is defined as Z = (X, Y) for each of the N samples of two random variables X & Y. For each point $z_i = (x_i, y_i)$, its neighbours are ordered according to $d_z ij = max\{||x_i - x_j||, ||y_i - y_j||\}$. Let d_k be the distance to the kth neighbor of z_i according to the metric above. Then the number of points x_j's $(j \neq i)$ within d_k distance to x_i is denoted as n_x, and the analogous definition for n_y. The

estimator then averages the value of the digamma function of number of samples of x_j's and y_j's which lie within the distance of the kth neighbour of z_i i.e. n_x and n_y, over all N samples of z_i and the final form of the estimator is as follows, with ψ being the digamma function and $<>$ indicating the averaging operator:

$$\widehat{I}(X;Y) = \psi(k) + \psi(N) - < \psi(n_x + 1) + \psi(n_y + 1) > \quad (4)$$

III. Evaluation of Estimators

To understand the effects of scaling and dimensionality on the estimator, the performances nearest neighbour and the kernel density estimators were compared on synthetic datasets.

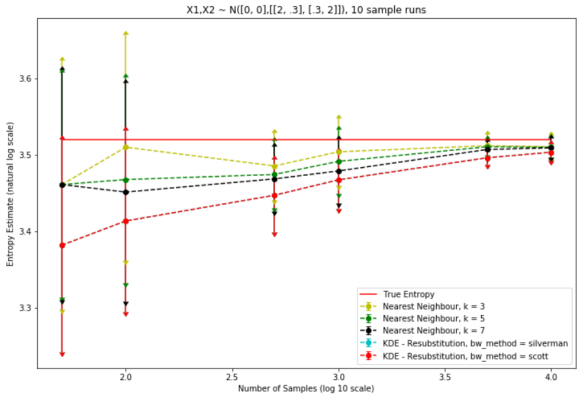

Fig. 1. Nearest Neighbour vs Kernel Density: Bi-Normal Distribution

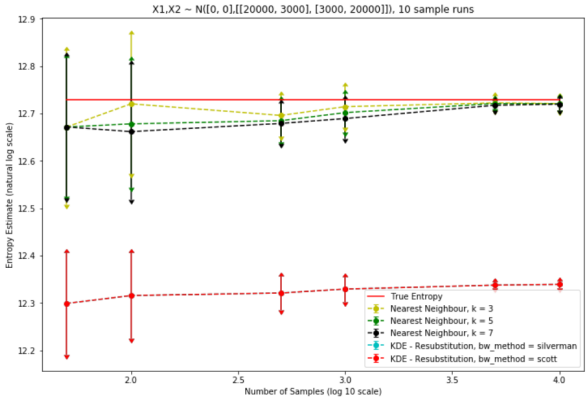

Fig. 2. Nearest Neighbour vs Kernel Density: Scaled Bi-Normal Distribution

A. Simulation Model

To compare the estimators, a standard bi-variate Gaussian distribution with a small correlation was used to generate the samples. To evaluate the effects of scaling, the co-variance matrix was multiplied by a factor of 10,000 and for dimensionality, the distribution was switched to four dimensional correlated Gaussian. For the nearest neighbour estimator, the values 3, 5 and 7 were taken for k. For the kernel density estimator, the kernel bandwidth was chosen according to either Scott's or Silverman's [5] rule of thumb, using Python library Scipy [6]. The experiment was repeated for 10 sample runs, and the mean and standard deviation were used to obtain the error-bar plots.

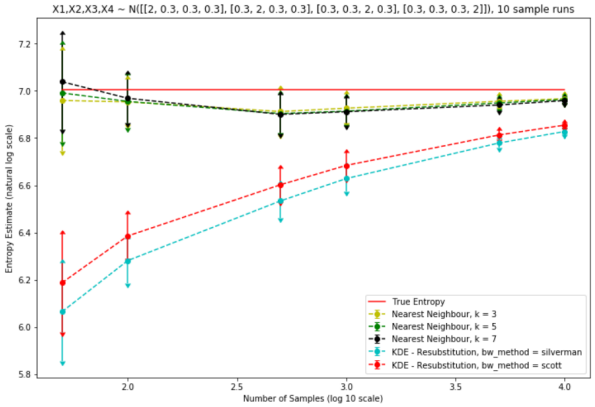

Fig. 3. Nearest Neighbour vs Kernel Density: Multivariate Normal Distribution

B. Results

The nearest neighbour estimator only slightly outperforms the kernel density estimation in case of the standard bi-variate distribution, while it does so significantly in the case of both scaled and multivariate distributions. Interestingly, the KDE consistently gives lower estimates for the scaled distribution, pointing to perhaps a biased model and possible improvement by changing kernel bandwidth. For the multivariate distribution, the KDE seems to asymptotically converge to the true value, but requires a far larger sample size to do so as compared to the nearest neighbour method. Similar trends are observed for the uniform distribution as well.
In conclusion, the K-NN estimator is a much more robust estimator and values of k$\sim$1-10 work well for most problem scenarios considered. It is therefore chosen as the de-facto estimator, unless specified otherwise.

IV. Optimally Picking the Last Coordinates

To obtain further insights about the structure of the solution that results in the optimal information bottleneck as per the nearest neighbor estimator, an experiment was conducted to pick the last or the last two samples using grid-search.
The samples were generated from the following distribution of random variables X, Y with U denoting the uniform distribution:

$$Y \sim U(0,2),\ Z \sim U(0,0.5)$$
$$X = Y + Z \mod 2$$

The setup above gives two correlated uniform random variables X and Y with same support. Besides the last or the last two, the remaining $\widetilde{X}$ points were sampled from $U(0,2)$. Due to the computational requirements for a fine grid-search, the total number of sampled points was set to 4. The experiments were repeated for multiple sample runs. For picking the last point, the five runs are presented in the same plot [Fig:4].
The experiment was repeated for picking the last two points with the variation of the IB value across the grid presented as a 2-D contour plot. The results obtained were consistent across the multiple sample runs, however the resulting plots

are omitted due to spacial constraints. Further, to check for consistency across different distributions, the picking of last two points experiment was repeated for the Gaussian distribution in Section:III, with $\widetilde{X} \sim N(0, 2)$ except for the last two samples.

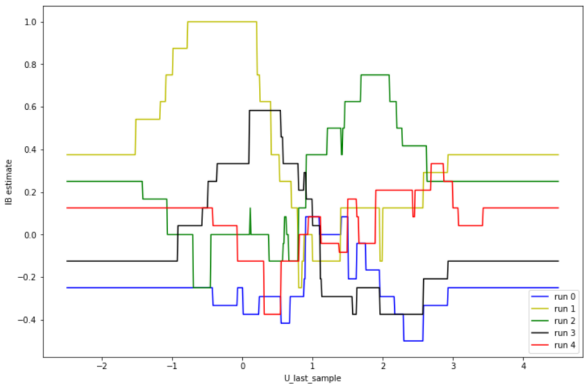

Fig. 4. Variation of the IB value with last sample: Uniform Distribution

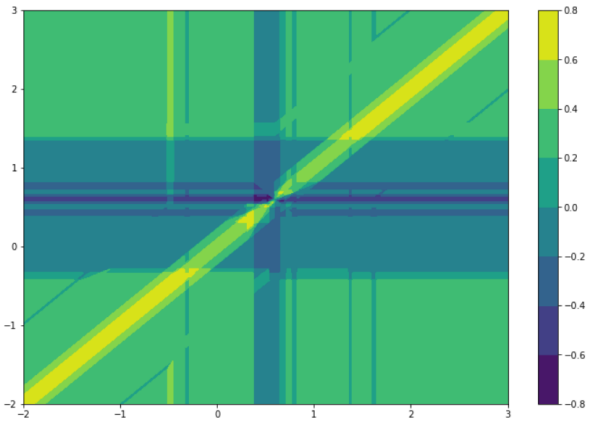

Fig. 5. Variation of the IB value with last 2 samples: Uniform Distribution [(X,Y) coordinates represent chosen samples, color bar indicates IB value]

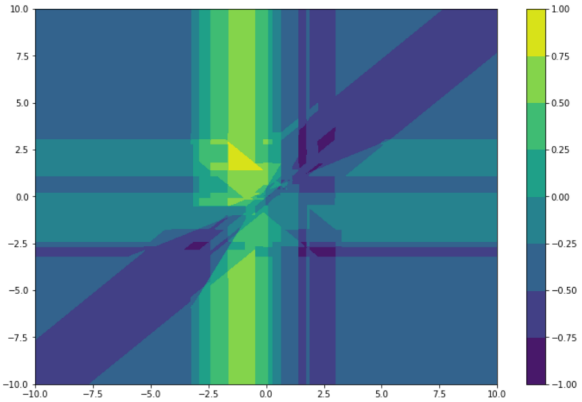

Fig. 6. Variation of the IB value with last 2 samples: Gaussian Distribution [(X,Y) coordinates represent chosen samples, color bar indicates IB value]

A glaring observation about the plots obtained for both paradigms (keeping aside the grid resolution) is the seemingly non continuous nature of the estimated values of the Information Bottleneck by the nearest neighbour estimator, as evidenced by the sharp changes. The second aspect is the non-uniqueness of the solution, as evidenced by the partitioning of the contour plot.
It is important to remember that this is a gross simplification of the original problem, which involves picking all N samples. The plots do however help us to appreciate the difficulty of optimizing the IB value and attempting to reach the minima in this particular formulation.

V. Co-ordinate Descent

Based on the previous experiment's findings, it seems that the complexity of the algorithm to solve for the set of samples which achieve the optimal Information bottleneck must be exponential in N. It is possible to bypass this constraint by developing a smooth approximation of the estimated Information bottleneck value, where the optimization is now over N samples and the hat denotes the fact that the optimization is with respect to the nearest neighbor information estimator:

$$\min_{\widetilde{X}_1..\widetilde{X}_N} \widehat{L} = \widehat{I}(\widetilde{X}; X) - \beta \widehat{I}(\widetilde{X}; Y) \tag{5}$$

A. Assumptions

Some assumptions are made about the estimator and information bottleneck for the purposes of simplifying the algorithm:

1) The nearest neighbour estimator has $k = 1$, that is it considers only the closest nearest neighbour.
2) β is set to 1 for the setup.

The first assumption slightly reduces the quality of the estimator, but greatly simplifies the algorithm. The second assumption produces a simpler closed form expression and nevertheless makes for an interesting test case. Both assumptions are non binding and may be removed, but are retained for this particular study for the sake of simplicity.

B. Notations & Setup

$N = 25$ samples are obtained from jointly Gaussian and positively correlated random variables (X, Y). The representation to be learnt is denoted by the random variable U and its corresponding N samples $U_1....U_N$. The joint variable $Z = (U, X)$ whereas $W = (U, Y)$.

C. Derivation

First consider the loss function:

$$\begin{aligned} \widehat{L} &= I(U; X) - \beta I(U; Y) \\ \textit{Setting } \beta \textit{ to } 1 &: I(U; X) - I(U; Y) \\ &= H(UX) - H(UY) \\ &= H(Z) - H(W) \end{aligned}$$

The k-nearest neighbor estimator for the joint entropy is of the form:

$$H(X) = \frac{1}{N-1}\sum_{i=1}^{N} log(\epsilon_X(i)) + C(k,N) \tag{6}$$

Here $\epsilon_X(i)$ denotes the distance of X_i from its kth neighbour, while C is a constant depending solely upon values of k & N. Plugging this in the previous expression gives us the following objective function:

$$\min_{U_1..U_N} log\,(\prod_{i=1}^{N} \frac{\epsilon_W(i)}{\epsilon_Z(i)}\,) \tag{7}$$

Substituting $\epsilon_W(i)$ with $\max\{\epsilon_X(i), \epsilon_Y(i)\}$ and vice versa for $\epsilon_Z(i)$, as per our chosen information estimator:

$$\min_{U_1..U_N} log\,(\prod_{i=1}^{N} \frac{\max\{||Y_i - Y_{ik}||, ||U_i - U_{ik}||\}}{\max\{||X_i - X_{ik}||, ||U_i - U_{ik}||}\,) \tag{8}$$

The above expression, however much simplified, is still not suited for optimization purposes due to the non-differentiable nature of the maximum function. To rectify this, the max function is replaced by the log of sum of exponentials. The final function is as follows:

$$\min_{U_1..U_N} log(\prod_{i=1}^{N} \frac{log(exp(\alpha||Y_i - Y_{ik}||) + exp(\alpha||U_i - U_{ik}||)}{log(exp(\alpha||X_i - X_{ik}||) + exp(\alpha||U_i - U_{ik}||)}) \tag{9}$$

With some large value of α, the above function is a reasonable approximation for our original objective and being differentiable, it may be optimized using standard gradient descent. Some other points of note:

- An $N * N$ responsibility matrix is maintained to indicate the nearest neighbour for each point, thus reducing the complexity of each gradient calculation.
- The responsibilities are updated after one complete set of updates for the complete set of points, based on the assumption that the nearest neighbors don't change mid update.
- For this particular implementation, the step size decays with each set of updates. Moreover, there is no explicit stopping criteria defined for the algorithm yet. Consequently, the number of iterations are fixed beforehand.

While the objective function approximates the estimated Information bottleneck (5) fairly well as evident in Fig:7, it is fairly susceptible to changes in initialization. Different initial samples drawn from the same distribution can converge to vastly different information bottleneck values and distributions altogether. This behaviour makes sense from an information theoretic point of view when considering the equivalence of the two markov chains, $(Y \leftarrow X \leftarrow \widetilde{X} \leftarrow f(\widetilde{X}))$ & $(Y \leftarrow X \leftarrow f(\widetilde{X}) \leftarrow \widetilde{X})$, whenever f is a $1-1$ mapping.

VI. Histogram Estimator

The Information Bottleneck is further explored but in a geometrically simpler binary paradigm. This necessitates as switch to the simpler histogram based estimator, given the small support set. The objective remains the same, finding a sequence of $\widetilde{X}_i$'s to minimize the information bottleneck (5). Only this time the underlying estimator is the histogram estimator. It first creates an estimate for the marginal and the joint distributions from the available samples, and then plugs them into the K.L. Divergence formula for Information:

$$\widehat{I}(X:Y) = D(\widehat{p}(x,y)||\widehat{p}(x) * \widehat{p}(y)) \tag{10}$$

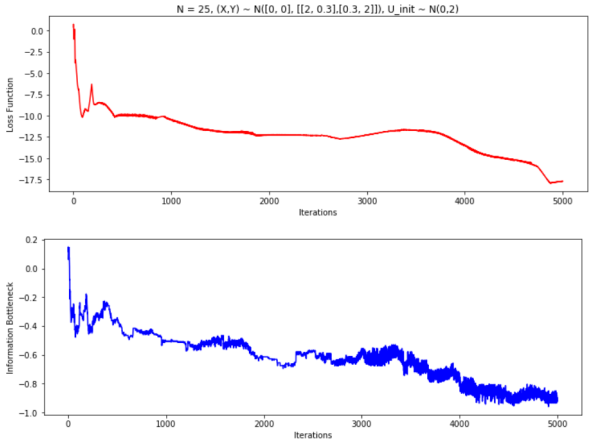

Fig. 7. Co-ordinate Descent: Comparing Loss & Information Bottleneck

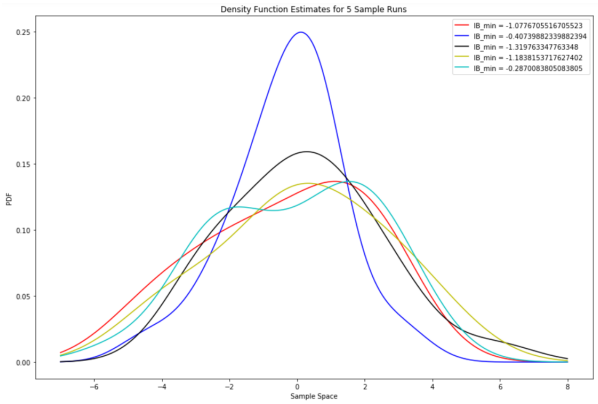

Fig. 8. PDF Visualization & Corresponding IB_min values: 5 Sample Runs

A. Model

Random variable $Y \sim B(\frac{1}{2})$ and X is obtained from Y as the output of a Binary symmetric channel with flipping probability $\alpha = 0.2$. It is important to note that in this particular paradigm, the choice of beta is very important, because it represents how much the function values deviation from X against the similarity to Y.

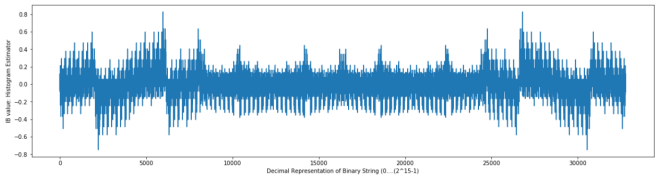

Fig. 9. IB values for all possible $\widetilde{X}_i$ sequences

B. Approach

Brute-forcing the way to the solution for $\beta = 1$ yields the optimal $\widetilde{X}_i$'s as either equal to the sequence $Y_1..Y_N$ or the sequence exactly complementary to $Y_1..Y_N$. In fact, the values of Information Bottleneck for different sequences of $\widetilde{X}_i$'s is symmetric in complementary sequences (Fig:9).
The second approach is to construct $\widetilde{X}$ as the output of a Binary symmetric channel with input X with flipping probability γ. The goal is to then find an optimum γ to minimize the bottleneck. The optimal solution is also possible by just plotting the IB values for different values of γ, which allows us to appreciate the role of β.

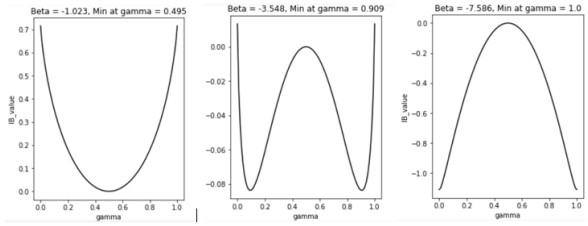

Fig. 10. IB Curve variation with β values

Since the maxima and minima for the two information terms occur at the same spot, the curve changes from convex to concave upon increasing β, in the fashion observed in Fig:10. The same results are reproducible with simulations for sufficiently large sample sizes, where the value of Information Bottleneck is calculated by estimating the joint distribution from the samples.

VII. Conclusion & Future Scope

There is an inherent non-uniqueness in the question which searches for the optimal information bottleneck. This non-uniqueness manifests itself in both continuous and discrete paradigms, independent of the type of estimator employed. The nearest neighbor estimator allows for a differentiable approximation. A more detailed study of the behaviour of this algorithm and the characteristics of the solutions obtained under different initializations or distributions could potentially offer useful insights or perhaps even allow us to sidestep the inherent non-uniqueness of the Information Bottleneck problem. Another aspect is the value of β and how it controls the entire nature of the solution, as seen in the last section, and therefore may also prove to be of interest under different problem scenarios.

Acknowledgements

The author would like to thank Prof. Michael Gastpar for offering the chance to attempt such as study, as well as his crucial guidance. The author would also thank Inovan Reka for his helpful suggestions.

References

[1] Tishby, Naftali, Fernando C. Pereira, and William Bialek. "The Information Bottleneck Method." ArXiv:Physics/0004057, April 24, 2000. http://arxiv.org/abs/physics/0004057.
[2] Beirlant, Jan, Edward J. Dudewicz, László Györfi, and Istvan Denes. "Nonparametric Entropy Estimation. An Overview," 1997.
[3] "L. F. Kozachenko, N. N. Leonenko, 'Sample Estimate of the Entropy of a Random Vector', Probl. Peredachi Inf., 23:2 (1987), 9–16; Problems Inform. Transmission, 23:2 (1987), 95–101." Accessed January 8, 2020.
[4] Kraskov, Alexander, Harald Stoegbauer, and Peter Grassberger. "Estimating Mutual Information." Physical Review E 69, no. 6 (June 23, 2004): 066138. https://doi.org/10.1103/PhysRevE.69.066138.
[5] B. W. Silverman. Density Estimation for Statistics and Data Analysis. Chapman & Hall/CRC, 1998.
[6] Eric Jones and Travis Oliphant and Pearu Peterson and others, Title = SciPy: Open source scientific tools for Python, Year = 2001–, URL = "http://www.scipy.org/"