APPLICATION OF THE INFORMATION-THEORETIC CLUSTERING TO THE DISCRETE CHOICE

Alexandre Monti

Supervised by Pavel Ilinov, Evangelos Paschalidis and Professor Michel Bierlaire

Contents

1 Introduction

1.1 Basics of information theory

Before diving into different clustering algorithms using methods of information theory, we need to define the important concepts that we will use later on this paper.

Definition : Let X be a discrete random variable taking values in A_X with distribution p. The entropy of X is given by

$$H(X) = \sum_{x \in A_X} p(x) \cdot \log\left(\frac{1}{p(x)}\right)$$

with the convention that if $p(x) = 0$, then $0 \cdot \log\left(\frac{1}{0}\right) = 0$. The unit for the entropy is the bit.

The entropy captures the average information content (or amount of surprise) associated with the outcomes of a random variable. The entropy is maximum when the distribution is uniform and is zero if an outcome happens with probability 1.

Definition : Let X and Y be two discrete random variables taking values in A_X and A_Y respectively and with joint distribution p. The joint entropy of X and Y is given by

$$H(X,Y) = \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \log\left(\frac{1}{p(x,y)}\right)$$

One property of the joint entropy is that $H(X) + H(Y) \geq H(X,Y)$ with equality if and only if $X \perp\!\!\!\perp Y$. The intuition behind the joint entropy is closely similar to the one for the entropy. It gives the average information content associated with the outcomes of a random vector.

Definition : The conditional entropy of X given Y is

$$H(X \mid Y) = \sum_{y \in A_Y} p(y) \left[\sum_{x \in A_X} p(x \mid y) \log\left(\frac{1}{p(x \mid y)}\right)\right] = \sum_{x \in A_X, y \in A_Y} p(x,y) \log\left(\frac{1}{p(x \mid y)}\right)$$

The conditional entropy quantifies the average uncertainty about x when y is known.

We can link these different definitions together by the following property :

$$H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$$

Furthermore, $H(X \mid Y) \leq H(X)$.

We could ask ourselves : is there a way to measure how much information I can learn about one variable by observing the other ? This is exactly the question that mutual information answers.

Definition : The mutual information between X and Y is given by

$$I(X;Y) = H(X) - H(X \mid Y)$$

With the properties about conditional and joint entropy from above, we know that $I(X;Y) \geq 0$ and $I(X;Y) = I(Y;X)$. Indeed,

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X) + H(Y) - H(X) - H(Y \mid X) \\ &= H(Y) - H(Y \mid X) = I(Y;X) \end{aligned}$$

Remark : We can generalize all of the above to continuous random variables easily by replacing sums with integrals. In the continuous case, the entropy can be infinitely large and positive or negative.

Another interesting property of the mutual information is that we can directly link it to the Kullback-Leibler divergence. We first recall what the Kullback-Leibler divergence is :

Definition : Let p and q be two discrete probability distributions defined on the same sample space X. The Kullback-Leibler divergence (or relative entropy) from q to p is defined by

$$D_{KL}(p \mid\mid q) = \sum_{x \in X} p(x) \log \left(\frac{p(x)}{q(x)} \right)$$

Now, if we go back to the definition of mutual information, we have that

$$\begin{aligned} I(X;Y) &= H(X) - H(X \mid Y) \\ &= \sum_{x \in A_X} p(x) \cdot \log \left(\frac{1}{p(x)} \right) - \sum_{x \in A_X, y \in A_Y} p(x,y) \log \left(\frac{1}{p(x \mid y)} \right) \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \log \left(\frac{1}{p(x)} \right) - \sum_{x \in A_X, y \in A_Y} p(x,y) \log \left(\frac{1}{p(x \mid y)} \right) \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \left[\log \left(\frac{1}{p(x)} \right) - \log \left(\frac{1}{p(x \mid y)} \right) \right] \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \log \left(\frac{p(x \mid y)}{p(x)} \right) \\ &= \sum_{x \in A_X, y \in A_Y} p(x,y) \cdot \log \left(\frac{p(x,y)}{p(x) \cdot p(y)} \right) \\ &= D_{KL}(p(x,y) \mid\mid p(x) \cdot p(y)) \end{aligned}$$

Thus, we have that $I(X,Y) = D_{KL}(p(x,y) \mid\mid p(x) \cdot p(y))$.

With these few definitions, we are now ready to delve into the information-theoretic clustering methods that will be the main topic of this work.

1.2 Information Bottleneck method (IB)

Developed by Naftali Tishby, Fernando C. Pereira, and William Bialek in 1999 [1], the information bottleneck method is a powerful framework in machine learning and information theory that aims to extract important information from data while eliminating any irrelevant or duplicated elements. More precisely, the goal of this method is to find a concise representation of the input data X while preserving the relevant information that X gives about the output data Y.

To this aim, we need to solve the following optimization problem : let T be the compressed input data. Given the joint distribution $p(x, y)$, the optimized encoding distribution $q(t|x)$ corresponds to

$$\begin{aligned}\min_{q(t|x)} L[q(t|x)] &= I(X;T) - \beta I(T;Y) \\ &= \sum_{x,t} p(x,t) \cdot \log\left(\frac{p(x,t)}{p(x)p(t)}\right) - \beta \sum_{t,y} p(t,y) \cdot \log\left(\frac{p(t,y)}{p(t)p(y)}\right) \\ &= \sum_{x,t} q(t|x)p(x) \cdot \log\left(\frac{q(t|x)}{q(t)}\right) - \beta \sum_{t,y} q(y|t)q(t) \cdot \log\left(\frac{q(y|t)}{p(y)}\right)\end{aligned}$$

with the additional Markov constraint $T \longleftrightarrow X \longleftrightarrow Y$. This constraint ensures that T can only have information about Y through X.

The idea behind this optimization problem is that the first term pushes for compression, while the second term emphasizes the importance of keeping the relevant information. In this context, β serves as the Lagrange multiplier associated with the constraint of retaining meaningful information. We denote by p the fixed distributions and by q the distributions that we can change or choose.

By making variational calculus, we find that this problem has a formal solution given by :

$$\begin{aligned}q(t|x) &= \frac{q(t)}{Z(x,\beta)} \exp[-\beta D_{\mathrm{KL}}[p(y|x) \mid\mid q(y|t)]] \\ q(y|t) &= \frac{1}{q(t)} \sum_x q(t|x)p(x,y) \\ Z(x,\beta) &\equiv \exp\left[-\frac{\lambda(x)}{p(x)} - \beta \sum_y p(y|x) \log \frac{p(y|x)}{p(x)}\right]\end{aligned}$$

where $D_{\mathrm{KL}}(P(x) \mid Q(x)) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$ denotes the Kullback-Leibler divergence. The $\lambda(x)$ in the definition of $Z(x, \beta)$ is the Lagrange multiplier for the normalization of the conditional distributions $q(t \mid x)$ at each x.

This solution is only formal because the first two equations depend on each other which makes them impossible to compute. But with an iterative approach, we can compute a solution which converges to the formal solution. This iterative algorithm works as follows :

Choose some initial distribution $q^{(0)}(t \mid x)$. Compute

$$q^{(0)}(t) = \sum_x p(x) q^{(0)}(t \mid x) \quad \text{and} \quad q^{(0)}(y \mid t) = \frac{1}{q^{(0)}(t)} \sum_x p(x,y) q^{(0)}(t \mid x)$$

The n-th iteration of the algorithm is given by :

$$\begin{aligned}
d^{(n-1)}(x,t) &\equiv D_{\mathrm{KL}}\left[p(y \mid x) \mid q^{(n-1)}(y \mid t)\right] \\
q^{(n)}(t \mid x) &= \frac{q^{(n-1)}(t)}{Z(x,\beta)} \exp\left[-\beta d^{(n-1)}(x,t)\right] \\
q^{(n)}(t) &= \sum_x p(x) q^{(n)}(t \mid x) \\
q^{(n)}(y \mid t) &= \frac{1}{q^{(n)}(t)} \sum_x q^{(n)}(t \mid x) p(x,y)
\end{aligned}$$

1.3 Deterministic Information Bottleneck (DIB)

One of the issue of the IB method is that it produces soft clustering which means that a given input can have multiple outputs with given probabilities. The idea behind the Deterministic Information Bottleneck method (DIB) [2] is to produce hard clustering, i.e. one input has a unique output. To this aim, we modify the cost function of the IB method as follows :

$$\min_{q(t|x)} L[q(t|x)] = H(T) - \beta I(T;Y)$$

still with the same Markov constraint $T \longleftrightarrow X \longleftrightarrow Y$.

How can we interpret the difference between IB and DIB methods? The disparity arises from a modification in the first term where $I(X;T)$ shifts to $H(T)$. In the domain of channel coding, $I(X;T)$ represents compression and suggests a limitation on communication cost between X and T. However, by adopting a source coding perspective, we can substitute $I(X;T)$ with $H(T)$, which denotes the representational cost of T. We can also see the difference between the two methods by substracting one to another :

$$\begin{aligned}
L_{IB} - L_{DIB} &= I(X;T) - \beta I(T;Y) - H(T) + \beta I(T;Y) \\
&= I(X;T) - H(T) = I(T;X) - H(T) \\
&= H(T) - H(T \mid X) - H(T) \\
&= -H(T \mid X)
\end{aligned}$$

This simple calculation reveals that the IB method promotes stochasticity in the encoding distribution $q(t \mid x)$. Indeed, $H(T \mid X)$ represents the stochasticity in the mapping from X to T.

In order to find a solution, we cannot use the same calculus method as for the IB problem. Rather, we

define the following family of cost functions :

$$L_\alpha \equiv H(T) - \alpha H(T \mid X) - \beta I(T;Y)$$

It is trivial to see that $L_{IB} = L_1$. We could say the same for $L_{DIB} = L_0$ but we will use a different strategy in order to find a solution for the DIB method. Indeed, we define the DIB solution as

$$q_{DIB}(t \mid x) = \lim_{\alpha \to 0} q_\alpha(t \mid x)$$

We can now use the same variational approach as for the IB method to find a formal solution which is given by :

$$\begin{aligned}
d_\alpha(x,t) &\equiv D_{\mathrm{KL}}[p(y \mid x) \mid q_\alpha(y \mid t)] \\
l_{\alpha,\beta}(x,t) &\equiv \log(q_\alpha(t)) - \beta d_\alpha(x,t) \\
q_\alpha(t \mid x) &= \frac{1}{Z(x,\alpha,\beta)} \exp\left[\frac{1}{\alpha} l_{\alpha,\beta}(x,t)\right] \\
q_\alpha(y \mid t) &= \frac{1}{q_\alpha(t)} \sum_x q_\alpha(t \mid x) p(x,y)
\end{aligned}$$

where $Z(x,\alpha,\beta)$ is a normalization factor given by

$$\begin{aligned}
z(x,\alpha,\beta) &= \frac{1}{\alpha} - 1 + \frac{\lambda(x)}{\alpha p(x)} + \frac{\beta}{\alpha} \sum_y p(y \mid x) \log\left(\frac{p(y \mid x)}{p(y)}\right) \\
Z(x,\alpha,\beta) &= \exp[-z(x,\alpha,\beta)]
\end{aligned}$$

Now that we have a general formal solution, we can take the limit $\alpha \to 0$ to finally have a formal solution for the DIB problem. By computing the limit, we find that

$$\lim_{\alpha \to 0} q_\alpha(t \mid x) = f : X \to T$$

where

$$\begin{aligned}
f(x) \equiv t^* &= \arg\max_t [l(x,t)] \\
l(x,t) &\equiv \log(q(t)) - \beta D_{\mathrm{KL}}[p(y \mid x) \mid q(y \mid t)]
\end{aligned}$$

More explicitly, for a fixed x the limit of $q_\alpha(t \mid x)$ when $\alpha \to 0$ becomes a delta function which is 1 at the value of t which maximizes $l(x,t)$. This is why we call this method the deterministic information bottleneck as the solution is a deterministic encoding distribution. The term $\log(q(t))$ in the equation promotes minimizing the number of unique values for t, favoring assigning each x to a t that already has many other x values associated with it. Meanwhile, the KL divergence term, similar to the original IB problem, ensures that the chosen t values retain as much information from x about y as possible.

The parameter β has the same role as in the original problem and controls the balance between the importance placed on compression and prediction.

As before, the solution for the DIB problem is only formal and we need to use an iterative algorithm in order to be able to find a computable solution. The iterative algorithm works as follows.

Choose some initial distribution for $f^{(0)}(x)$. Set

$$q^{(0)}(t) = \sum_{x:f^{(0)}(x)=t} p(x) \quad \text{and} \quad q^{(0)}(y \mid t) = \frac{\sum_{x:f^{(0)}(x)=t} p(x,y)}{\sum_{x:f^{(0)}(x)=t} p(x)}$$

Then apply the following steps until convergence :

$$\begin{aligned}
d^{(n-1)}(x,t) &\equiv D_{\mathrm{KL}}\left[p(y \mid x) \mid q^{(n-1)}(y \mid t)\right] \\
l_\beta^{(n-1)}(x,t) &\equiv \log(q^{(n-1)}(t)) - \beta d^{(n-1)}(x,t) \\
f^{(n)}(x) &= \arg\max_t [l_\beta^{(n-1)}(x,t)] \\
q^{(n)}(t \mid x) &= \delta\left(t - f^{(n)}(x)\right) \\
q^{(n)}(t) &= \sum_x p(x) q^{(n)}(t \mid x) = \sum_{x:f^{(n)}(x)=t} p(x) \\
q^{(n)}(y \mid t) &= \frac{1}{q^{(n)}(t)} \sum_x q^{(n)}(t \mid x) p(x,y) = \frac{\sum_{x:f^{(n)}(x)=t} p(x,y)}{\sum_{x:f^{(n)}(x)=t} p(x)}
\end{aligned}$$

The intuition behind this algorithm is the following:

- Calculate the Kullback-Leibler (KL) divergence between the true conditional distribution $p(y \mid x)$ and the current estimated conditional distribution $q^{(n-1)}(y \mid t)$. The KL divergence tells us how much $p(y \mid x)$ is "close" to $q^{(n-1)}(y \mid t)$. If $D_{KL} = 0$, the two distributions are the same.
- Calculate $l_\beta^{(n-1)}(x,t) = \log(q^{(n-1)}(t)) - \beta \cdot d^{(n-1)}(x,t)$, which combines compression and relevance terms using the Lagrange multiplier β.
- Assign each data point x to the cluster t that maximizes $l_\beta^{(n-1)}(x,t)$.
- Update the conditional distribution of clusters given data points $q^{(n)}(t \mid x)$ by setting it to a Dirac delta function $\delta(t - f^{(n)}(x))$.
- Update the marginal distribution of clusters $q^{(n)}(t)$ by summing over all data points assigned to each cluster.
- Update the conditional distribution of data points given clusters $q^{(n)}(y \mid t)$ by computing the ratio of the joint distribution $p(x,y)$ to the marginal distribution $p(x)$ for each cluster.

2 Implementation of the IB and DIB methods

2.1 Geometric DIB algorithm

We first implement the DIB method for geometric clustering to have a good insight of what is going on.

We consider two dimensions data points $\{\mathbf{x}_i\}_{i=1:n}$ and we want to cluster them. Here, we need to choose what is X and what is Y. We first make the same choice as in [3] where X is the data point index i and Y is the data point location $\mathbf{x}$. Thus, we want to cluster data indices i in a way that keeps as much information about data location as possible.

We need now to choose the joint distribution $p(i, \mathbf{x}) = p(i)p(\mathbf{x} \mid i)$. It is trivial to choose $p(i) = \frac{1}{n}$ but the choice for $p(\mathbf{x} \mid i)$ is more complicated. Following [3], we take

$$p(\mathbf{x} \mid i) \propto \exp\left[-\frac{1}{2s^2} d(\mathbf{x}, \mathbf{x}_i)\right]$$

where s corresponds to the units of distance and d is a distance metric, for example the Euclidean distance $d(\mathbf{x}, \mathbf{x}_i) = ||\mathbf{x} - \mathbf{x}_i||^2$.

2.2 General IB algorithm

We now implement the general information bottleneck algorithm based on Algorithm 1 of [2]. The objective is to determine whether we can replicate the IB curve in Figure 1 outlined in the paper.

By making multiple tests, we understand that the only condition for β is to be greater than 0 and not in the interval $[0; 1]$ as we thought before. With this observation, we achieve to make an IB curve :

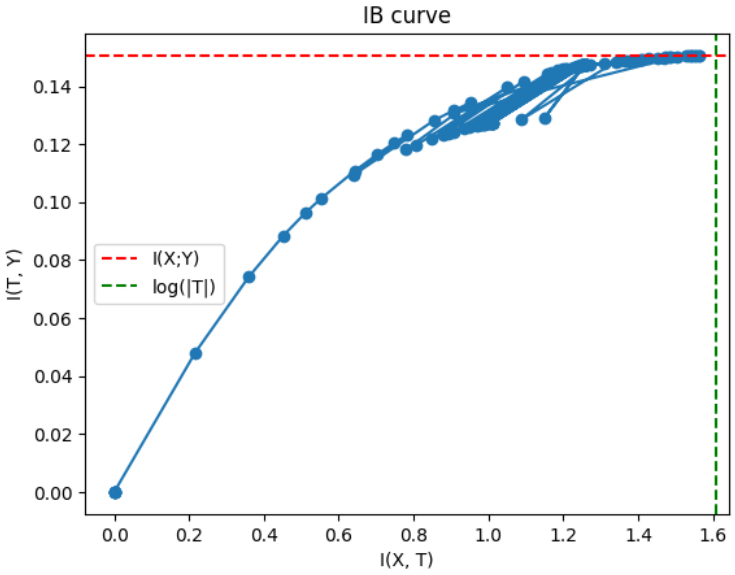

Figure 1: IB curve for a random joint distribution with $|X| = 5$, $|Y| = 3$ and $\beta \in [0; 200]$. We fix the number of iterations for each β to 1000.

The purpose of an IB curve is to illustrate the balance between compression, represented by $I(X;T)$, and prediction, denoted by $I(T;Y)$. Indeed, at the bottom left of the curve, maximizing compression (minimizing $I(X;T)$) prioritizes reducing redundancy and extracting the most essential information from the input data X but we see that by doing this, we lose almost all information about Y and prediction is pretty bad. Conversely, at the other end, maximizing prediction (maximizing $I(T;Y)$) emphasizes capturing as much relevant information as possible to accurately predict the output Y given the latent variable T. By doing this, we see on the curve that we are forced to make little compression.

The bounds represented by a red horizontal line and a green vertical lines are the theoretical bounds for $I(T;Y)$ and $I(X;T)$. $I(T;Y)$ is bounded by $I(X,Y)$ because $I(X;Y)$ is the mutual information if we do not compress input data. On the other hand, $I(X;T)$ is bounded by $\log(|T|)$ because

$$I(X;T) = H(T) - H(T \mid X) \leq H(T) \leq \log(|T|)$$

We can prove that $H(T) \leq \log(|T|)$ easily.

Proposition : Let X be a discrete random variable with distribution p. Then $H(X) \leq \log(|X|)$ with equality if and only if p is the uniform distribution.

Proof.

$$
\begin{aligned}
H(X) &= \sum_x p(x) \cdot \log\left(\frac{1}{p(x)}\right) = \mathbb{E}\left[\log\left(\frac{1}{p(X)}\right)\right] \\
&\leq \log\left(\mathbb{E}\left[\frac{1}{p(X)}\right]\right) \quad \text{by Jensen's inequality for concave functions, here } \log(x). \\
&= \log\left(\sum_x p(x) \cdot \frac{1}{p(x)}\right) = \log(|X|)
\end{aligned}
$$

Thus, $H(X) \leq \log(|X|)$. Furthermore, we know that Jensen's inequality[1]is an equality if and only if ϕ is affine or if X is constant. Here, $\phi(x) = \log(x)$ which is clearly not affine so $H(X) = \log(|X|) \iff p(X)$ is constant, i.e. p is the uniform distribution. □

Let's go back to the IB curve. Upon closer inspection, we notice certain points lying below the curve. These points correspond to β values where the algorithm may have struggled to converge, likely due to insufficient iterations. Indeed, setting the number of iterations to 1000 may not always ensure convergence of the algorithm.

Thus, two questions remain :

- What is the optimal number of iterations in order to ensure the convergence of the algorithm ?
- Which value of β gives the best tradeoff between compression and prediction ? In other words, which β is at the crossroads of the red and the green curves ?

2.2.1 Convergence of the algorithm

Instead of fixing the number of iterations, we could choose a metric that ensures convergence. For example, we can compare two consecutive distributions $q(t)$ and stop the algorithm when the difference between these two distributions is lower than a certain threshold. More precisely, if $q^{(n)}(t)$ is the marginal distribution of t at the n-th step, we can iterate the algorithm while $||q^{(n)}(t) - q^{(n-1)}(t)|| > \theta$ where θ is a chosen threshold. By doing this, we can achieve better results as shown in the plots below :

[1]Jensen's inequality for concave functions : $\mathbb{E}[\phi(X)] \leq \phi(\mathbb{E}[X])$.

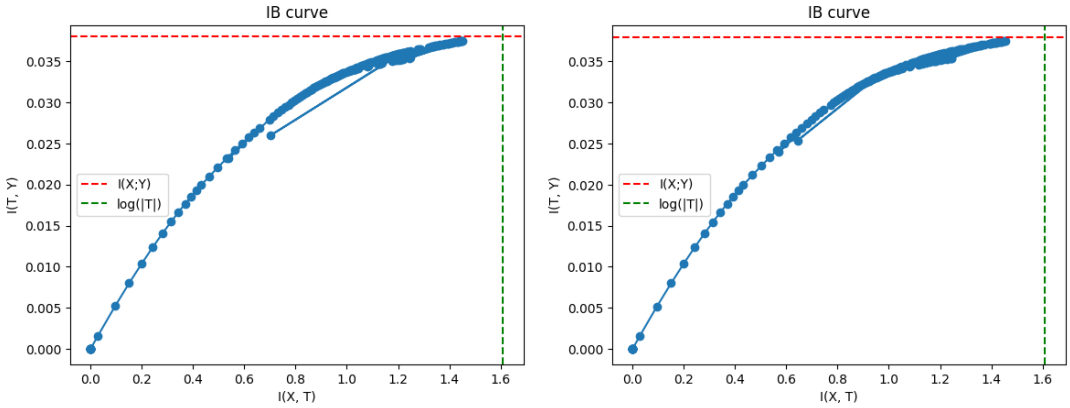

Figure 2: Comparison of the IB curve with a fixed number of iterations for each β (left) and the IB curve using the metric to ensure convergence (right). We fixed the number of iterations to 100 and the threshold θ to 1e-8.

We can clearly see that the algorithm using the metric to ensure convergence has indeed better performance. The curve is smoother and less points are below the curve. We can still see some points under the curve but this comes from the fact that we add a maximum number of iterations in the algorithm to decrease computation time. We fixed this maximum number of iterations to 10'000 which means that we could retrieve the right curve with the first algorithm if we set the number of iterations to 10'000.

2.2.2 Optimal β

The question of determining the ideal β value, where we intersect the red and green curves on the IB curve, is not quite meaningful. It typically leads to a large β value that diminishes compression, essentially negating the utility of the variable T.

Instead, the research of an optimal β relies on the preferred quantity of clusters. The IB curve illustrates the upper bounds of prediction performance achievable when the number of clusters, represented by $I(X;T)$, is fixed. Consequently, this research depends on the specific goals of our analysis.

2.3 TO DO LIST

- Essayer de comprendre pouquoi l'algo 2 pour le geometric clustering fonctionne pas
- Implémenter le DIB général

2.4 Questions about implementation

I have a problem with implementation. Suppose we have 4 points $(0,0)$, $(1,1)$, $(5,5)$ and $(6.5,6.5)$. We take $f^{(0)}(x_i) = i$ as in the paper [02] page 5 "For the DIB, we initialized the cluster assignments

to be as even across the cluster as possible, i.e. each data points belonged to its own cluster". At the initialization, $p(y \mid x_i) = q^{(0)}(y \mid i)$. So $d^{(0)}(x_i, t) = 0$ when $i = t$ and $d^{(0)}(x_i, t) > 0$ otherwise. But then, $l_\beta^{(0)}(x_i, t)$ is always the biggest when $i = t$ as $\log(q(t)) < 0$ for $\beta > 0$. This means that the algorithm do not move from the initial clustering which assigns each point to its own cluster. I think I am missing something about this step...

3 References in APA format

[1] Tishby, N., Pereira, F. C., & Bialek, W. (2000). The information bottleneck method. arXiv preprint physics/0004057.

[2] Strouse, D. J., & Schwab, D. J. (2017). The deterministic information bottleneck. Neural computation, 29(6), 1611-1630.

[3] Strouse, D. J., & Schwab, D. J. (2019). The information bottleneck and geometric clustering. Neural computation, 31(3), 596-612.

[4] MacKay, D. J. (2003). Information theory, inference and learning algorithms. Cambridge university press.