Information Theory (2021/22)
Homework 3: Estimating Entropies with ITE

Tommaso Bergamasco (ID: 2052409)

Contents

1 Introduction

In this Homework I'm going to replicate the procedures described in Homework 2, by using the *Information Theoretical Estimators (ITE) in Python* toolbox.

2 Estimating the Differential Entropy $h(x)$

ITE provides 6 different ways to estimate the differential entropy $h(x)$ of a vector x.

The generic syntax to compute an estimation is as follows:

```
co1 = ite.cost.cost_name()
entropy = co1.estimation(x)
```

I therefore replicated this 2 lines of codes for the 6 estimators methodologies which were:

1. k-nearest neighbors.
2. Approximate slope of the inverse distribution function.
3. Maximum entropy distribution (method 1).
4. Maximum entropy distribution (method 2).
5. -KL divergence from the normal distribution (M).

6. -KL divergence from the uniform distribution (M).

Where notice that the first 4 estimators are what are called "**Base**" estimators (namely based on the definition), while the last 2 (denoted by M) are the so called "**Meta**" estimators (namely the ones derived from other estimations).

2.1 Test on x Gaussian

In Figure 1 and Figure 2 we can observe the quality of the estimations (from HW2 and from ITE) w.r.t. the true value. First I vary the variance P of the gaussian distribution (we know that its mean m_x doesn't change anything from the statistical point of view), then I vary also the length of the vector x on which the estimations are performed.

Comments and observations are reported in the captions.

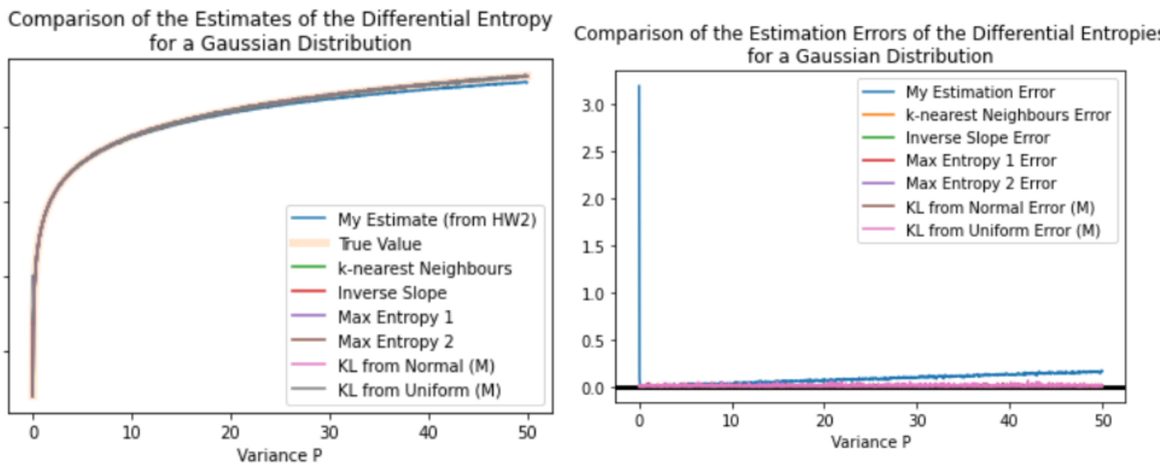

Figure 1: Estimation of the differential entropy in the univariate gaussian case (left) and estimation errors w.r.t. the true value computed in HW2 (right).
All the estimations behave nicely except for the one computed in HW2 which diverge from the true value for high values of the variance (for which we already commented in the past homeworks that the estimation becomes harder).
On the other hand the ITE estimators seem to behave very well even in this case.
Note also the error peak regarding my estimation for very small variances, which is instead perfectly handled by ITE.

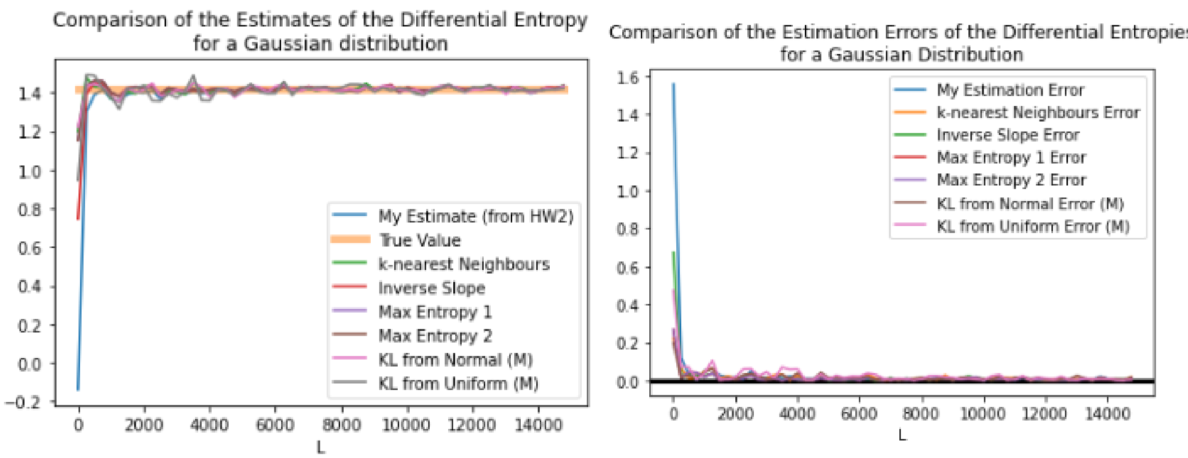

Figure 2: As the above image but varying the length L of the vector x instead.
All the estimators converge pretty fast to the true value but we can also observe that my estimator has the largest error when L is not big "enough".

2.2 Test on x Exponential

In Figure 3 and Figure 4 we do the same as above for an exponential distribution.

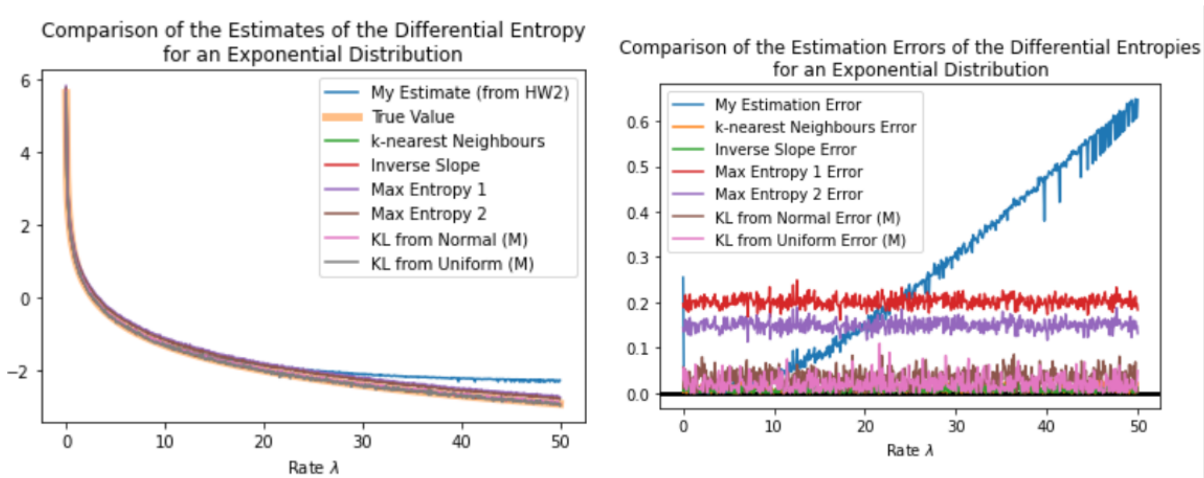

Figure 3: Comparison of the estimates (from HW2 and ITE) for an exponential distribution with varying rate λ.
Note how my estimate deviates significantly from the true value when the rate λ increases (namely when the exponential pdf is more concentrated around 0^+). Note also the error peak of my estimate for small λ (which is due to the fact that the exponential pdf is in this case "heavy-tailed" and would require $L >>$ to perform accurate estimations).
Note how the majority of ITE estimates behaves very well w.r.t. the true value, while the 2 methods based on the "maximum entropy principle" show a small offset, which anyway seems invariant to the value of λ.

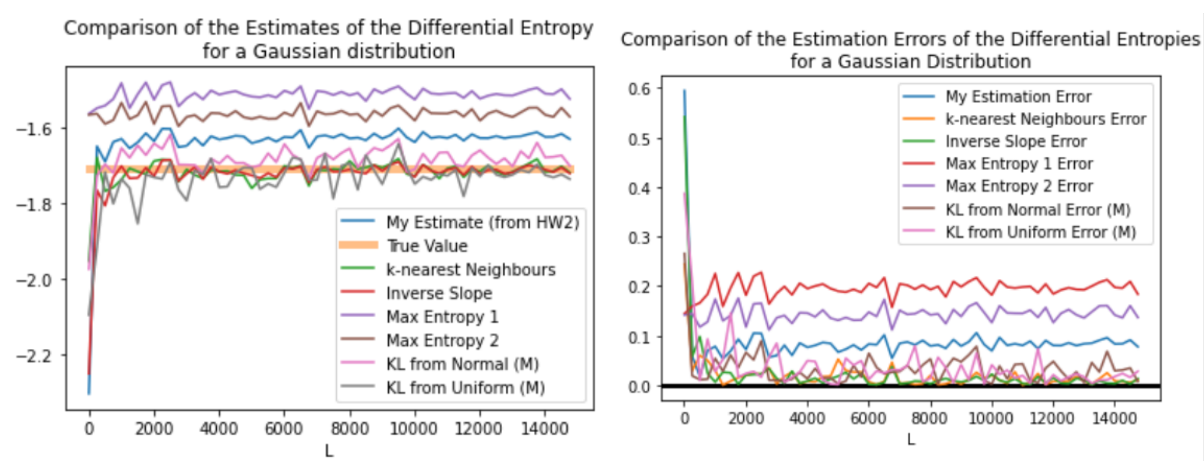

Figure 4: Same as Figure 3 but now I'm varying L.
Note once again how the majority of the ITE estimators work well, while my estimator and the ones based on the "maximum entropy principle" are the ones which deviate the most from the true value.
By watching the right side of the figure we can also see how the estimators which are less robust to small L are respectively: estimator from HW2, the "Inverse Slope", and the "Meta" estimators derived from the uniform and normal distribution.

3 Estimating the Joint Quantities

When dealing with more than one vector the toolbox provides a restricted number of estimating techniques, which have all been used in the homework.

In particular:

- For the joint entropy $h(x, y)$ I used the first estimator used to estimate the Shannon entropy (k-neighbors) on a matrix with 2 columns (this was inspired by the matlab documentation since in the python documentation are not shown estimators for this quantity).
- For the conditional entropy $h(x|y)$ was available only one estimator based on the reduction to the Shannon entropy principle.
- For the relative entropy were available three estimators: two of them based on the k-neighbors principle (base estimators) and the other one obtained as the difference between the cross entropy and the entropy (meta estimator).
- For the mutual information $I(x; y)$ were available two meta estimators one derived from the relative entropy and the other based on the Shannon and joint entropies of the vectors.

3.1 Test on x (Gaussian) and y

In Figure 5 we can see the joint quantities estimations in the gaussian case.

From the figure it's possible to see that my estimation in HW2 was significantly deviated from the actual value.

Performing some experiments I located the reason for this bad behaviour in the choice of the bins' width. In facts in HW2 I forced the same width (precision) both for the x and y distributions. The plotted results are already refined by giving to Python 2 different precisions for the 2 distributions, but could probably get even better if some sort of adaptive choice (w.r.t. the variance σ_x^2) of the width is used. The original error in the joint entropy estimation is then propagated in the estimation regarding the conditional entropy and the mutual information.

Regarding the relative entropy note how my estimation is not computed when σ_x^2 starts to grow. This is due to the fact that when $\sigma_x^2 >> \sigma_z^2$ then the ratio $p_x/p_y \to \infty$. Note also how, instead, the ITE estimator is able to perform the estimation even for $\sigma_x^2 >>$.

Finally note that we observe here for the first time the phenomenon for which not every ITE estimator is able to compute an actual estimation and in facts some of them are missing from the plots (e.g. see relative entropy or mutual information).

This strange behaviour is signaled by a warning given by the ITE library, which indicates that a division by 0 has occured. Such warning seems to be input-dependent as we will see in the rest of the experiments.

4 Test on single ECG Signals

I now test the procedure described in Section 2 by estimating the entropy of the 3 ECG signals provided for the homework (see Figure 6).

As for the gaussian multivariate case we can note that some ITE estimators are missing from the plots even if the exact same procedure seen in Section 2 have been used.

The missing estimators are:

- The one base estimator based on the k-nearest neighbors principle
- The 2 meta estimators.

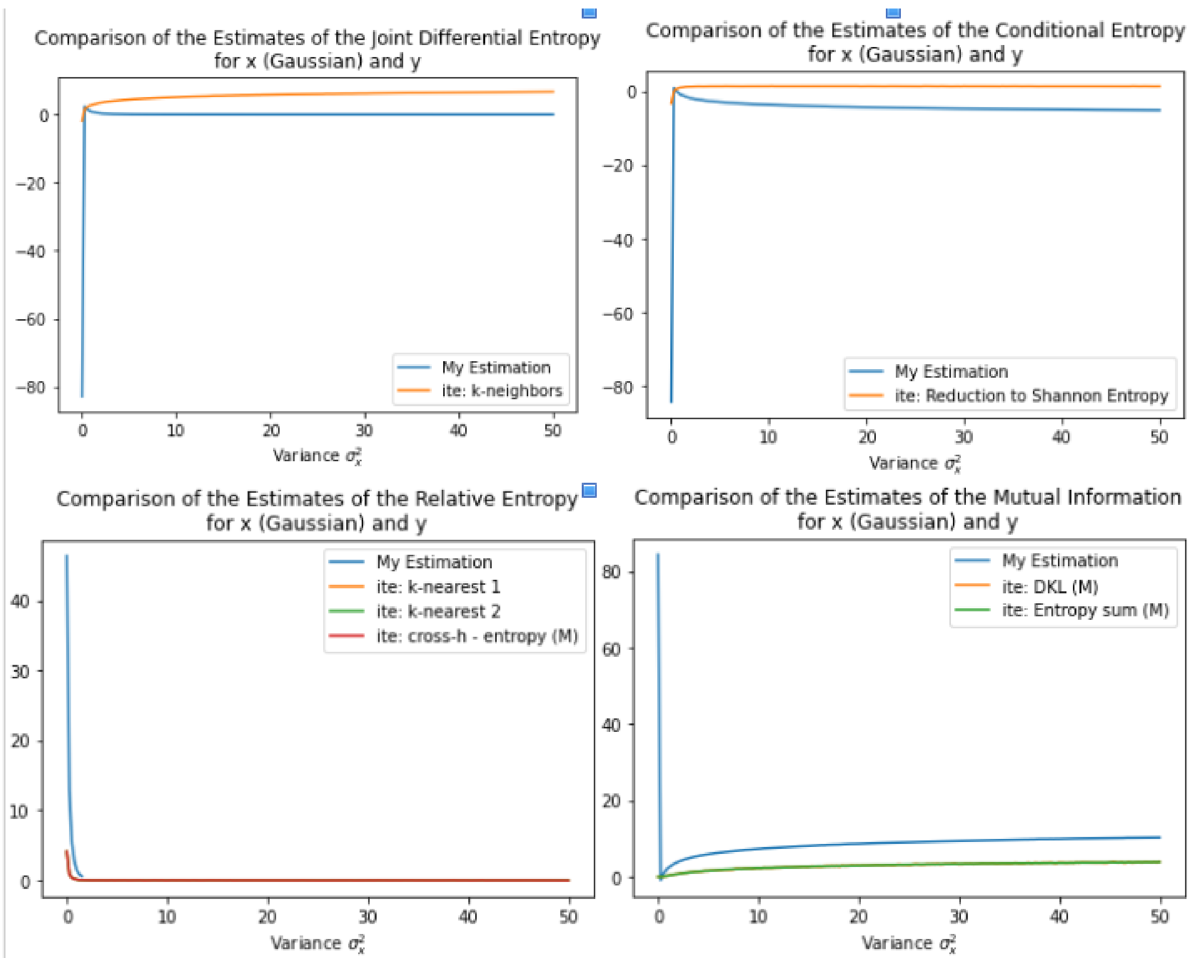

Figure 5: Joint Quantities for x Gaussian and $y = ax + bz$ (z gaussian).

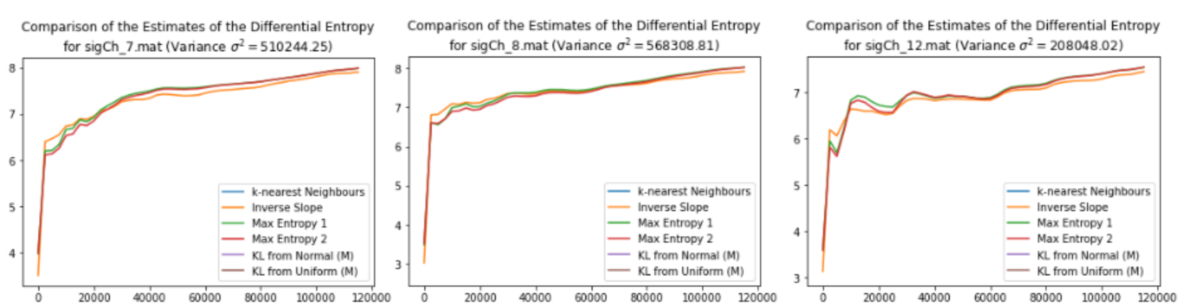

Figure 6: Differential Entropy estimation for the 3 ECG signals provided for the homework. The estimation is performed on the first L samples of each vector (L variable).

Note that:

- The estimated differential entropy tends to grow with the number of considered samples L.
- The last signal has the most "noisy estimation" since it has a variance ($\sigma_{12}^2 \approx 200000$) which is 4 times the variance of the 2 first signals ($\sigma_7^2 \approx \sigma_8^2 \approx 50000$).

5 Test on couples of ECG Signals

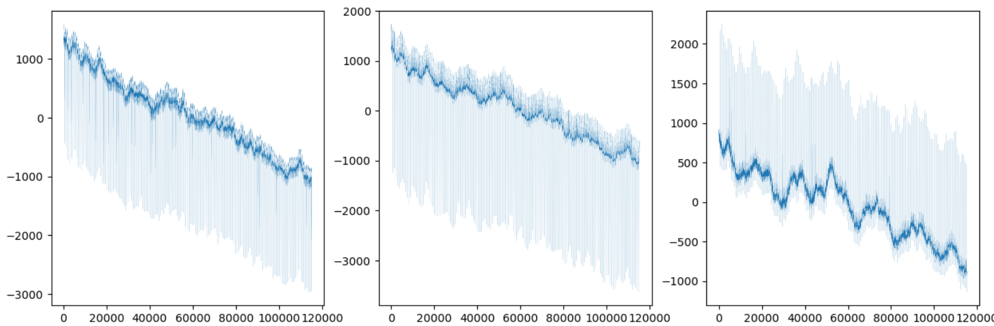

Figure 7: The 3 ECG signals. From left to right `sigCh_7.mat`, `sigCh_8.mat` and `sigCh_12.mat`

In Figure 8 are shown the estimated quantities regarding the signal "7" and "8". As we can see ITE is not estimating any of the required quantities except for the relative entropy for which only 1/3 of the estimations are performed.

As we have seen for the differential entropy of the single ECG signals even in this case the estimations seem to be input-dependent, in fact by applying the same procedure also to the other 2 couples of signals the output changes.

In Figure 9 and Figure 10 is possible to see the estimated quantities respectively for the couple "8-12" and "7-12".

In both cases we get:

- The estimation of the joint entropy.
- No estimation of the conditional entropy.
- 1/3 estimations of the relative entropy.
- 1/2 estimations of the mutual information.

Note also that, the joint entropy and the mutual information estimations stop respectively for $L \approx 10000$ and $L \approx 20000$. On the other hand the relative entropy is not estimated for $L > 20000$ in Figure 9, while in Figure 8 and Figure 10 it is estimated even for $L = L_{max} = 115200$.

Regarding the estimated quantities note the following in Figure 9 and 10:

- The joint entropy increases with L.

In facts by looking at Figure 7 we can note that every signal has an "overall negative slope". This means that by increasing L, we are likely increasing also the alphabet of the 2 r.v. which are virtually generating the ECG signals, leading to an increased joint entropy.

- The relative entropy decreases with L.

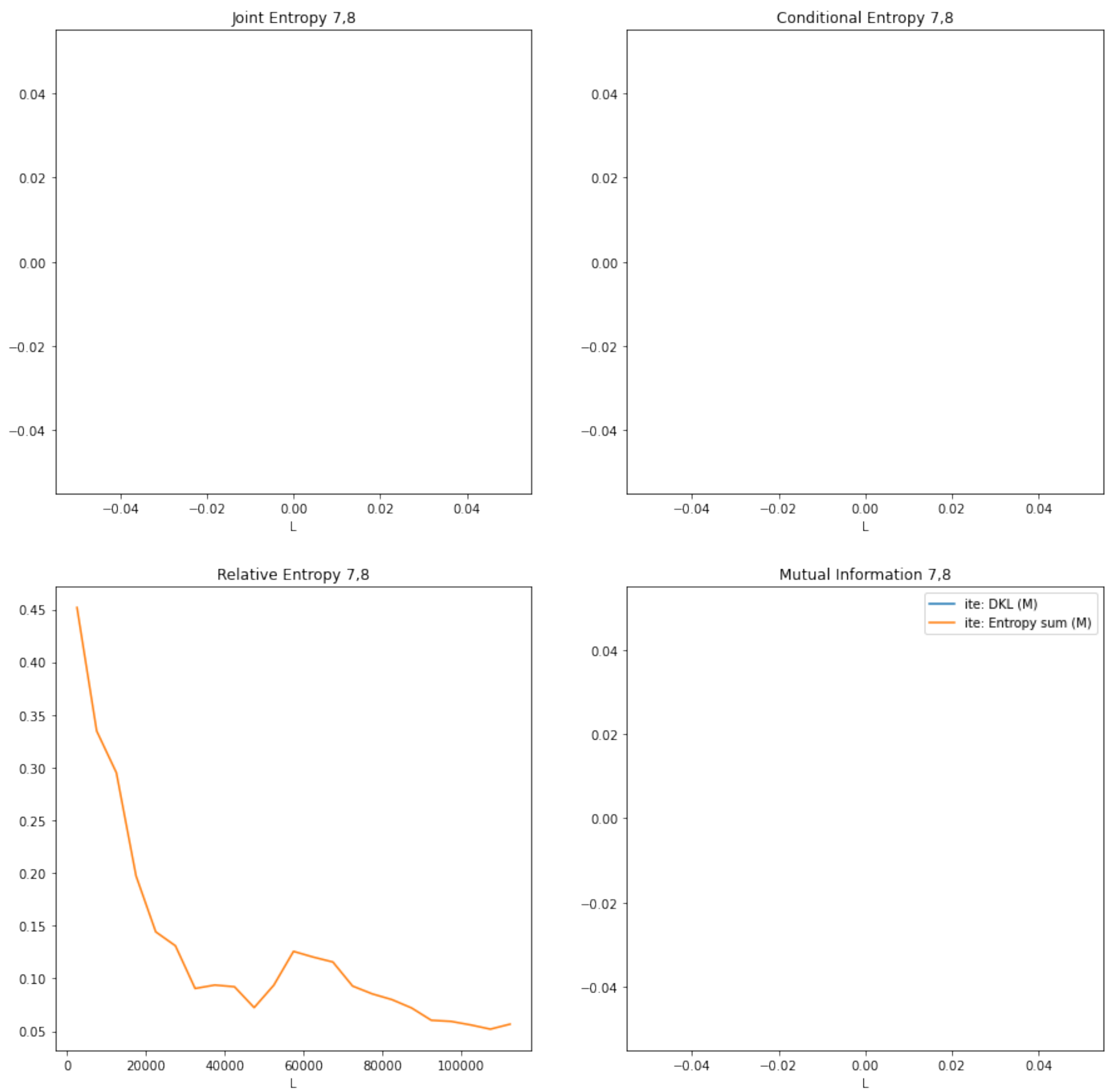

Figure 8: Estimated quantities for the `sigCh_7.mat` and `sigCh_8.mat` signals. Except for 1 of the relative entropy estimators, the ITE library is computing not even a single value for each sub-vector obtained by considering only the first L samples (where L varies).

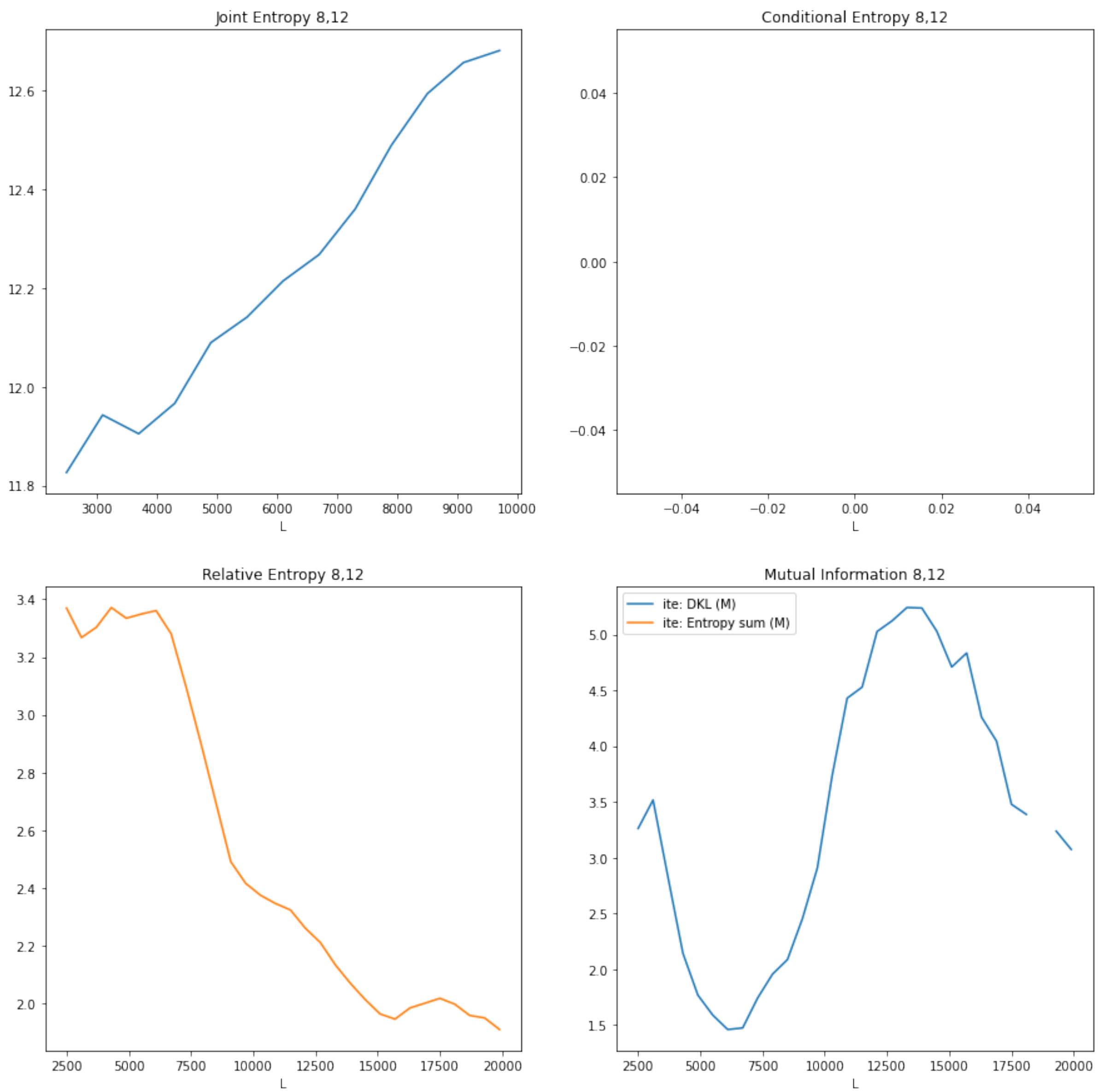

Figure 9: Estimated quantitites for `sigCh_8.mat` and `sigCh_12.mat`. As above we can note that the majority of estimators is not computed. Anyway this time ITE is able to compute the joint entropy and $1/2$ of the mutual information estimations.

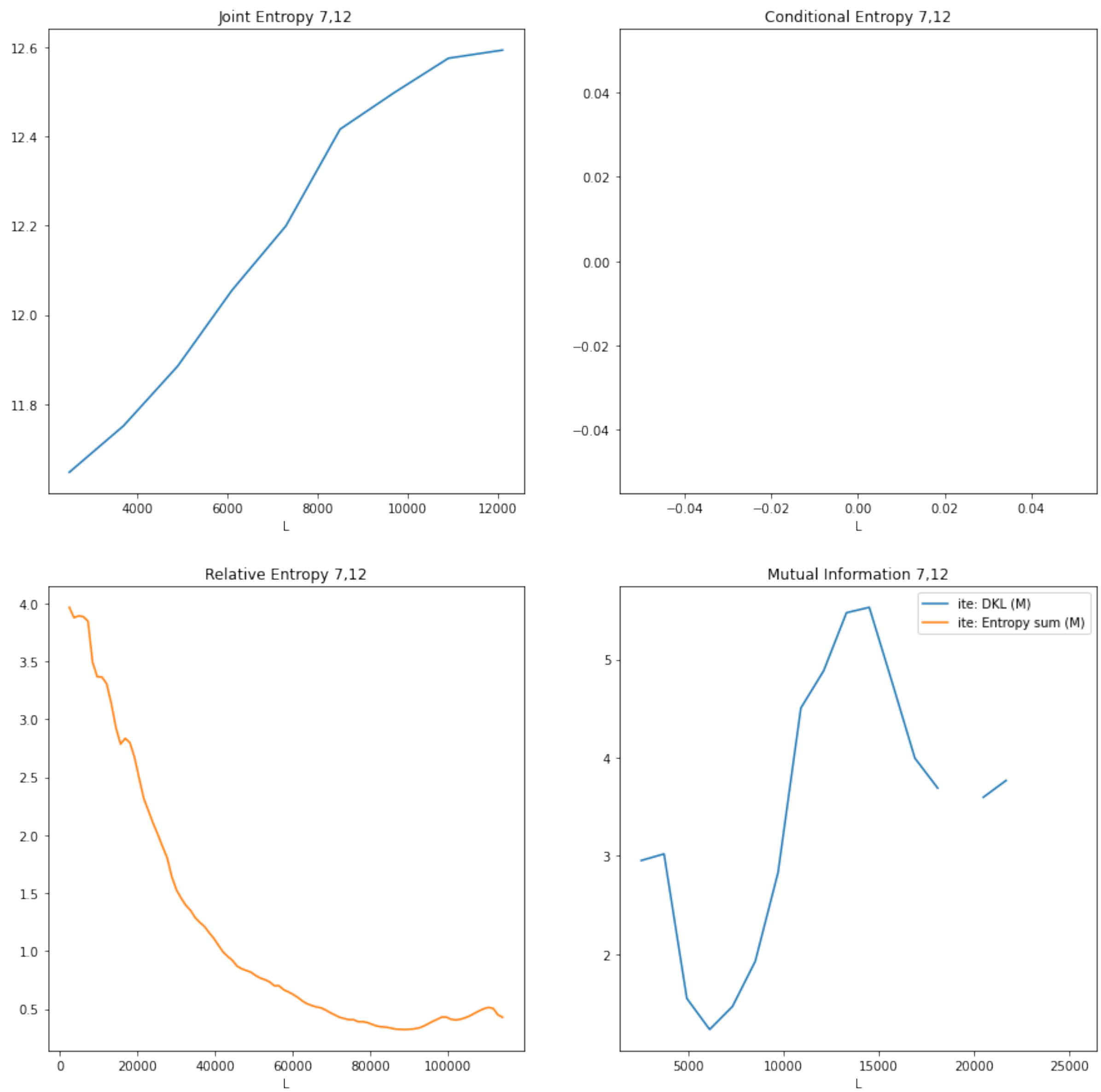

Figure 10: Bonus: estimated quantities for `sigCh_7.mat` and `sigCh_12.mat`. The observed output is similar to the one of Figure 9.

In facts since the relative entropy is a measure of statistical distance the more samples from the 2 vectors we are considering the more we are able to see (also from an intuitve point of view) that the 2 signals belong to the same "class" of biological signals. For small values of L the estimation is instead deceived by the local characteristics of the signals.

- The mutual information has an oscillatory behaviour.

This is probably due to the "macro-oscillatory" behaviour of the ECG signals which is possible to observe in Figure 7. Anyway as already observed, the mutual information is estimated only for $L < 20000$ and hence, we could expect $I(x;y)$ to stabilize for $L \uparrow$.

6 Entropy Rates

On the online documentation regarding the ECG signals we find out that "each signal is digitized at 1000 samples per second". Therefore to estimate the entropy rates for the 3 ECG signals we can use the following formula:

$$\text{Entropy Rate} = FH_s(x) = 1000 \cdot H_s(x)$$

where:

$$H_s(x) = \text{Average Entropy for Symbol} = \frac{1}{n} H(\boldsymbol{x}).$$

As estimator for the entropy i choosed the one based on the inverse slope principle, the results are the following:

- Entropy rate of ECG7: 0.068623
- Entropy rate of ECG8: 0.068765
- Entropy rate of ECG12: 0.064674

Note that the lowest entropy rate is achieved by the last signal, which has (as we already observed above) the highest variance.

Since both the entropy rate and the variance can be seen as indicators of the complexity of a signal, this result could seem counterintuitive, but is due to the fact that the entropy rate measures not only how wide is the range of values that the signal can take, but also how every simbol is correlated to the others.

This tells us that for the last signal the symbols' correlation is higher than the 2 first ECG signals or, equivalently, that given n symbols of the ECG12 it's easiest to predict the next symbols.

7 Bibliography

Szabo, Zoltan. 'Information Theoretical Estimators Toolbox'. arXiv, 8 May 2014. https://doi.org/10.48550/arXiv.1405.2106.