Analyse et Traitement de l'Information

TP6: Entropy and Detection Theory

1 Quantifiers of information

In information theory, the amount of information present in a random variable (r.v.) is quantified with **entropy** which is a function, only of the probability mass function. Intuitively, entropy indicated how surprising the outcome of an experiment, or equivalently the observed value of an r.v. would be.

Mutual information, on the other hand, specifies the amount of information between two r.v's. In other words, it indicates how much the knowledge of one r.v. helps in predicting the value of another r.v.

Using the definitions in the lecture notes, try to solve the below problem. Pay careful attention to the differences between marginal, conditional and joint versions of entropy and mutual information.

U, V and W are three binary random variables. Their joint probability mass function is depicted as below:

U	V	W	$p(u, v, w)$
0	0	0	$\frac{1}{4}$
0	0	1	0
0	1	0	$\frac{1}{4}$
0	1	1	$\frac{1}{8}$
1	0	0	0
1	0	1	$\frac{1}{8}$
1	1	0	0
1	1	1	$\frac{1}{4}$

Try to calculate the information measures below:

a $H(U)$, $H(V)$ and $H(W)$

b $H(U|V)$, $H(V|U)$ and $H(W|U)$

c $I(U;V)$, $I(U;W)$, $I(U;V,W)$

d $H(U,V,W)$

For each of the above items you could directly use the definition. However, because they are related to each other in many ways, you could calculate some of them and derive the rest by using their relations. Chain rules for entropy and mutual information and drawing the Venn diagram could especially be helpful.

2 Source coding

Source coding tries to express the output of a probabilistic source of information as compact as possible. In fact this problem is very much connected to the problem of data compression.

The idea is to capture the redundancies in a source of information based on the probabilities of the occurrences of different outcomes. Intuitively speaking, source coding tries to assign shorter codes for outcomes that are more probable and longer codewords for less probable events. As a result, the overall code-length assigned to the whole sequence will be ideally as small as possible. In fact, this smallest possible value is closely related to the entropy.

Imagine a very simple case where we have a memoryless source X that generates a sequence $X_1, X_2, \cdots, X_n$ of random variables. The probability distribution of all these X_i's are identical and independent from each other (*i.i.d.*). The source is binary and produces '1's with probability $\theta = 0.1$ and '0's with probability $1 - \theta = 0.9$.

Let's consider that the alphabet is constituted of all possible sequences $X_1, X_2, \cdots, X_5$ of length 5 (the size of this alphabet is then $2^5 = 32$).

i. Generate this sequence with the length $n = 10,000$. You have now generated a text of 2000 symbols by considering all the consecutive subsequences of length 5.

ii. One very simple source coder is the Huffman code. Install the Python package `pip install dahuffman`, or use the MATLAB commands '*huffmandict*' and '*huffmanenco*' to encode the binary sequence above.

iii. Calculate the entropy of this sequence analytically and compare it with the length of the output of the above encoder. Can you comment on this? Do they correspond? You may divide by the length of the sequence to give an interpretation.

Binary Hypothesis Testing

1 **General idea:**

In a simple hypothesis testing scenario, suppose we want to detect the presence of the constant-valued signal $A = 1$ corrupted in a white Gaussian noise of zero mean and unit variance. Suppose we want to have the probability of miss (P_M) and probability of false acceptance (P_{FA}) to have equal values.

a. Formulate the problem in a hypothesis testing setup, i.e., indicate the null and alternative hypotheses ($\mathcal{H}_0$ and $\mathcal{H}_1$) and the detection rule for the Bayesian strategy when both hypotheses have the same probability.

b. Find the minimum required samples for the detector to have $P_M = P_{FA} \leq 0.001$. To analytically derive the results, you could use the approximation for the Q-function, the tables or the relevant functions in MATLAB / Python /... [see slide 12]

2 **A communication systems example:**

Imagine a simple communication system where you send a string of bits to a channel. The bits are simply modulated as a voltage signal of amplitude one when the bit value is '1' and of amplitude zero when the bit value is '0'. The duration for each of these states is fixed to N samples. The channel receives this voltage signal and adds a zero-mean AWGN of σ_z^2 to it to create the output. The aim is to estimate the input of the channel, based on the value we observe at the output.

Now we want to formulate this communication systems problem as a detection problem and link it to the previous question. In particular, at each interval (of length N), we want to see if the input was '0' or '1'. In other words, we want to detect the presence of the constant-valued signal $A = 1$ in the output.

To realize this idea, generate a random binary sequence $\{b_i\}_{i=1}^{k}$ of length $k = 10000$ and its respective voltage signal as explained above. Use the value N that you calculated in the previous problem as the duration of each interval for each bit. Therefore, the signal you send to the channel as the input, say $x[n]$ should be a pulse wave with a total length kN. Add a zero mean Gaussian noise $z[n]$ of variance $\sigma_z^2 = 1$ to $x[n]$ to generate the output of the channel $y[n]$.

a. Having generated $y[n]$, try to estimate $x[n]$ based on the optimal detection rule you know and then restore the original binary sequence $\{b_i\}_{i=1}^{k}$ from $\hat{x}[n]$

b. Plot $x[n]$, $y[n]$ and $\hat{x}[n]$ on the same figure and zoom-in the figure appropriately.

c. Compare $\{b_i\}_{i=1}^{k}$ with its estimate $\{\hat{b}_i\}_{i=1}^{k}$ and calculate P_M, P_{FA} and $P_e = P_M + P_{FA}$ and compare them with the values you expect from the analytic setup of the first question.

3 **Neyman-Pearson approach:**

There is given a Gaussian signal $\mathcal{N}(\mu, \sigma_x^2)$ (choose signal parameters by yourself) corrupted in a white Gaussian noise of zero mean and variance σ_z^2.

a. Formulate the problem in a hypothesis testing setup, i.e., indicate the null and alternative hypothesis ($\mathcal{H}_0$ and $\mathcal{H}_1$) and the detection rule according to Neyman-Pearson criteria.

b. Plot a ROC curve (probability of detection vs energy-to-noise ratio) for different values of P_{FA} and explain the relations between the detection accuracy, length of the signal and variance of the noise. [see slide 25]

Multiple Hypothesis Testing *

In this part, we want to realize a simple identification system and use the idea of M-ary detection to identify a given noisy query. Suppose we have M people or objects for identification. From these M items, we extract a sequence of feature, e.g., images of faces of individuals, DNA sequences, fingerprint images or any other feature that could potentially make distinctions between the items. These M feature sequences, now in the form of M vectors are stored in a database $\mathcal{X}$.

Suppose one of these M people, say the person indexed as w whose features are registered in $\mathcal{X}$ comes and claims his/her identity. The identification system, assuming that the person is already registered, should identify that person correctly, i.e., it should find the index $\hat{w}$, where $1 \leq \hat{w} \leq M$. So an error is declared if $\hat{w} \neq w$.

This problem, formulated as a multiple hypothesis testing problem, could be treated with the idea of matched filters. The assumption we make here is that these feature vectors are almost orthogonal.

To investigate this idea by simulation:

a. Generate $M = 200$ signals with length $n = 1000$ where each of the samples are independent realizations of a random variable $X \sim \mathcal{N}(0, 1)$. Therefore, $\mathcal{X}$ should be a matrix with 200 rows, each representing an item, and 1000 columns as features.

b. Suppose the queries are noisy version of the items. Generate a matrix $\mathcal{Y}$, the same size as $\mathcal{X}$ by adding zero-mean Gaussian noise signals with $\sigma_z^2 = 100$ to the rows of $\mathcal{X}$.

c. Using the idea of matched filters for multiple detection, i.e., finding the maximum similarity of a given query signal (rows of $\mathcal{Y}$) with the signals to be detected (rows of $\mathcal{X}$), try to identify the query items. Remember that in this case, finding the maximum similarity is equivalent to finding the minimum Euclidean distance. Therefore, for any row of $\mathcal{Y}$, try to find the row of $\mathcal{X}$ that has the minimum Euclidean distance with it.

d. Compute the probability of error and compare it with the formula presented in the lecture notes [see slide 45].

Assessment

Please archive your report and codes in "Prénom Nom.zip" (replace "Prénom" and "Nom" with your real name), and upload to "`Upload TPs/TP6 Detection Theory`" on `https://moodle.unige.ch` before Monday, December 7 2020, 23:59 PM. Note, the assessment is mainly based on your report, which should include your answers to all questions and the experimental results.