ENSSAT - Informatique 3
Module Théorie et Codage de l'Information

Lab sessions

Objective:

The objective of this series of labs is to learn the basics of data manipulation with information theoretic tools such as entropy and mutual information which are often used to characterize the complexity of datasets and their relationships.

After playing with binary random sequences (i.e. bit sequences) you will investigate the use of information theoretic tools for image data, in applications like image compression, feature selection, and image registration. An application to genomics is also proposed.

Programming language:

Python code (.py) or Jupyter notebook (.ipynb).

Deliverables:

If not written as notebooks, the expected deliverables comprise a report of the work done in each lab, as well as the corresponding Python codes.

Course assessment:

The evaluation of the labs accounts for 1/3 of the final grade.

Lab 1: Source coding

1. Write a program/function which estimates the entropy (in bits/symbol) of a random data vector `x` of arbitrary length taking values into any finite set. Take care of `NaN` values which may occur.

2. Check that a sufficiently long vector of random independent, equiprobable bits, has entropy H (close to) 1 bit.

3. Draw the variations of the entropy as a function of $\alpha = P[X=0]$.

4. Consider a discrete source *with memory* using the binary alphabet {0, 1}, sequentially producing time-indexed symbols $x_1, x_2, \ldots x_k$. To compute the entropy of such a source, accounting for the memory effect would require conditioning the entropy of the current symbol with respect to its past up to some order q:

$$H(x_k) = \sum_{\mathcal{A}} H\left(x_k \middle| x_{k-1}, x_{k-2}, \cdots, x_{k-q}\right) p\left(x_{k-1}, x_{k-2}, \cdots, x_{k-q}\right)$$

   where $\mathcal{A}$ is the set of all possible outputs of the random variables $\{x_{k-1}, x_{k-2}, \cdots, x_{k-q}\}$. We consider here the case of a first order memory effect $q = 1$, also called a *Markov chain*. A first order Markov chain is completely defined by its transition matrix:

$$\mathbf{\Pi} = \begin{pmatrix} p(x_k = 0|x_{k-1} = 0) & p(x_k = 1|x_{k-1} = 0) \\ p(x_k = 0|x_{k-1} = 1) & p(x_k = 1|x_{k-1} = 1) \end{pmatrix} = \begin{pmatrix} p_{00} & p_{10} \\ p_{01} & p_{11} \end{pmatrix}$$

   Moreover, the Markov chain is said to be *stationary* if the *a priori* probabilities $\{p(x_k)\}$ verify the condition $\mathbf{P\Pi} = \mathbf{P}$, with $\mathbf{P} = [p(x_k = 0) \quad p(x_k = 1)] = [p_0 \quad p_1]$.

   Assuming the stationarity of the Markov chain, calculate the *a priori* probabilities p_0 and p_1 as a function of p_{00} and p_{01}, and find the expression of the entropy $H(x_k)$ of this source with memory.

5. As an application, consider processing the image of a scanned text page[1], similarly to a fax system (taking for instance pixel value 1 for white and 0 for black). First estimate the transition matrix from the pixel values of the image. Then evaluate the above entropy $H(x_k)$ and compare it to both the entropy of the corresponding memoryless source, and to the average size (in bits/pixel) of the original image. Conclude.

6. Try to implement an appropriate source encoder for this kind of data. *Hint:* consider RLE.

[1] https://foad.univ-rennes1.fr/course/view.php?id=1008552

Lab 2: Lossy image compression

Note: this lab is inspired from [1].

1. Download the ex9Data.zip file from [1]. We shall first work with the small `nrows` x `ncols` = 128x128 color image *f*.

2. Apply the *k*-means algorithm implemented in Python (in Scikit-Learn package) or Matlab (`kmeans`) to *f* to obtain:

   a. a set of *k* color vectors (centroids), and
   b. a partition of the image pixels into clusters, and the corresponding segmentation.

   Each cluster centroid is the representative of the set of pixel color vectors that are the nearest to it than to any other centroid. The centroids are the *codewords* of the compression method, and the set of codewords is called the *codebook*.

   To do this, you will first need to reshape *f* as a standard data array of size [`nrows`* `ncols` `3`] (one row per pixel and one column for color component).

3. Choose *k*=10 for instance. Create an approximation image $\hat{f}$ by mapping the labels issued from the *k*-means algorithm to their respective codewords.

4. Calculate the distortion measure *D*(*k*) as the mean square error (MSE) between $\hat{f}$ and *f*.

   Estimate the coding rate *R*(*k*) as the number of bits per pixel that would be obtained by encoding the cluster labels with the run length encoding (RLE) approach. The RLE code is a 2-D array reporting in the first column the label values encountered in the label sequence and in the second column the number of successive counts of this value in this sequence. For instance, if the first labels are:

   [9 9 9 3 4 6 6] ,

   then the RLE code would be:

   [9 3 ; 3 1 ; 4 1 ; 6 2] .

5. Repeat the same operation for *k* ranging from 2 to 20, and report the evolution of the distortion *D*(*k*) as a function of the coding rate *R*(*k*). What can you observe?

6. Try to implement a *real* codec by writing to / reading from a binary file the information which is necessary to store and retrieve $\hat{f}$, i.e. the image size, the number of clusters *k*, the codebook issued from the *k*-means algorithm, the size of the RLE code and the RLE code itself. The codebook can be written in `float` format, whereas the RLE can be written in `uchar` format.

7. Compare the R-D curve obtained by this system with the estimated one obtained above. What is coarsely the increase in bits per pixels for a given distortion measure? How could the codec be improved?

[1] openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises/ex9/ex9.html

Lab 3: Mutual information and NMI calculation

1. Create a function which computes the mutual information between two equal size data vectors x and y, $I(x;y)$. For this you have to estimate the joint pdf $p(x,y)$ and the marginal pdf's $p(x)$ and $p(y)$ from the contingency matrix. In case of continuous distributions it can be interesting to smooth the pdf estimates thanks to (low-pass) filtering.

2. It can be more interesting to use the normalized mutual information, defined as:

$$NMI(x;y) = \frac{2\,I(x;y)}{H(x) + H(y)} \quad ,$$

   which has the property: $0 \leq NMI(x;y) \leq 1$.

3. Compute the mutual information and NMI between all pairs of color components of the image used in Lab2. Which pair of components share the least amount of mutual information / NMI ?

4. Compute the NMI between each color component of this image with the cluster labels issued from the *k*-means algorithm (here considered having a discrete distribution, so smoothing makes no sense). Report the values of the mutual information for each component as a function of *k*, and interpret the results.

5. Consider samples of 10 DNA sequences of different animals (Cow, Carp, Chicken, Human, Loach, Mouse, Rat, Seal, Whale, Frog)[2]. As above, compute the NMI between each pair of DNA sequence. It can be useful to convert the string array of each sequence to integer values. Analyze the matrix of pairwise NMI values. For instance, which animal shares the highest NMI with the Human?

[2] https://foad.univ-rennes1.fr/course/view.php?id=1008552

Lab 4: IT-based feature selection applied to hyperspectral image band selection

The objective of this lab is to create a program able to select the (ordered) set of spectral bands that best represent the diversity and complementarity of a hyperspectral image (also called *data cube*), thanks to information theory.

A well-known algorithm in IT feature selection is the *minimal-redundancy-maximal-relevance* (mRMR) method of Peng *et al*. [1] This method is based on the following criterion:

$$z_k = \arg\max_{x \in X \setminus \{z_1, z_2, \cdots, z_{k-1}\}} \left[I(y;x) - \frac{1}{k-1} \sum_{i=1}^{k-1} I(z_i;x) \right]$$

where z_i is the *i*-th selected feature and *y* is the target class vector.

1. Implement the mRMR method and apply it to the AVIRIS hyperspectral image 'IndianPineHSI' and its target label 'IndianPineHSI_gt' [2]. Give the sequence of the first *M*=20 selected spectral bands.

2. Implement a (supervised) kNN classifier (for instance 5NN); take a proportion of say, 50% training pixels per class at random, the remaining being used for testing.

3. Report the overall accuracies (OA) of classified pixels (normalized trace of the confusion matrix) as a function of the number of selected bands obtained by mRMR, from 1 to 20.

4. As a comparison, proceed with the same train/test sets and perform the kNN classification with the same set of 20 selected features, but now presented in a different (random) order.

[1] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, pp.1226-1238, 2005; http://home.penglab.com/papersall/docpdf/2005_TPAMI_FeaSel.pdf

[2] https://foad.univ-rennes1.fr/course/view.php?id=1008552