ELEN-0060: Information and Coding Theory

Project 1 - Information Measures

Maxime Goffart
180521

Olivier Joris
182113

Academic year 2021 - 2022

1 Implementation

1.1 Function `entropy`

We are using this mathematical formula:

$$\mathcal{H}(\mathcal{X}) = -\sum_{X_i} P(X_i) \log_2 P(X_i)$$

Our implementation is first filtering the negative $P(X_i)$ because the $\log(x)$ function is only defined for $x \in]0, +\infty[$. Then, it is performing the sum over all X_i using the `sum` method of the numpy library and return the result according to the above mathematical formula.

Intuitively, the entropy is measuring the expected amount of information provided when observing an event corresponding to one value of the random variable. The less probable an event is, the more the amount of information provided is when this event occurs.

1.2 Function `joint_entropy`

We are using this mathematical formula:

$$\mathcal{H}(\mathcal{X}, \mathcal{Y}) = -\sum_{X_i, Y_j} P(X_i \cap Y_j) \log_2 P(X_i \cap Y_j)$$

We notice that this joint entropy is equal to the previously defined entropy taking as argument the joint distribution of the two random variables instead of the marginal distribution of the single random variable.

It is why we are implementing this function by simply reshaping the joint probability distribution to a one dimensional array. In this way, we can just call the `entropy` function with this reshaped array as argument.

1.3 Function `conditional_entropy`

We are using this mathematical formula which has been demonstrated in the theoretical course:

$$\mathcal{H}(\mathcal{X}|\mathcal{Y}) = \mathcal{H}(\mathcal{X}, \mathcal{Y}) - \mathcal{H}(\mathcal{Y})$$

Our implementation consists thus in computing the marginal probability distribution $P_{\mathcal{Y}}$ by marginalizing the joint probability distribution P_{xy} over $\mathcal{X}$. This is done using the `sum` function of the numpy library using `axis = 0` as argument. Then, we just compute the result using the above mathematical formula and the previous defined functions.

An equivalent way to compute this property is to directly use the mathematical formula of the entropy:

$$\mathcal{H}(\mathcal{X}|\mathcal{Y}) = -\sum_{X_i, Y_j} P(X_i \cap Y_j) \log_2 P(X_i|Y_j)$$

1.4 Function `mutual_information`

We are using this mathematical formula which has been demonstrated in the theoretical course:

$$\mathcal{I}(\mathcal{X};\mathcal{Y}) = \mathcal{H}(\mathcal{X}) - \mathcal{H}(\mathcal{X}|\mathcal{Y})$$

Our implementation consists thus in computing the marginal probability distribution $P_{\mathcal{X}}$ by marginalizing the joint probability distribution P_{xy} over $\mathcal{Y}$. This is done using the `sum` function of the numpy library using `axis = 1` as argument. Then, we just compute the result using the above mathematical formula and the previous defined functions.

The mutual information between two discrete random variables measures the level of dependence between these random variables. Especially, if $\mathcal{I}(\mathcal{X};\mathcal{Y}) = 0$, $\mathcal{X}$ and $\mathcal{Y}$ are independent.

1.5 Functions `cond_joint_entropy` and `cond_mutual_information`

We are using these mathematical formulas which have been demonstrated in the theoretical course:

$$\mathcal{H}(\mathcal{X},\mathcal{Y}|\mathcal{Z}) = \mathcal{H}(\mathcal{X},\mathcal{Y},\mathcal{Z}) - \mathcal{H}(\mathcal{Z})$$

$$\mathcal{I}(\mathcal{X};\mathcal{Y}|\mathcal{Z}) = \mathcal{H}(\mathcal{X}|\mathcal{Z}) + \mathcal{H}(\mathcal{Y},\mathcal{Z}) - \mathcal{H}(\mathcal{X},\mathcal{Y},\mathcal{Z})$$

Our implementations consist thus in computing the right marginal probability distributions by marginalizing the joint probability distribution P_{xyz}. This is done as in the previous sections using the `axis` argument of the `sum` function of the numpy library. Then, we just compute the result using the above mathematical formulas and the previous defined functions.