Hand in problem 1 in Information Theory (EITN45)

VT 2, 2017

Hand in Problem 1

In this problem you should implement MATLAB functions for the information measures $H(X)$ and $I(X;Y)$. The task will force you to understand the derivation of the functions. It will also be good to have the function during the rest of the course.

Entropy

The first function to be implemented is the entropy function. The function header should be specified as below, where `<Your code>` is a place holder for your code.

Entropy.m (File header)

```
function H=Entropy(P)
% The Entropy function H(X)
%
% P column vector: the vector is the probability distribution.
% P matrix: Each column vector is a probability distribution
% P scalar: The binary entropy function of [P; 1-P]
% P row vector: Each position gives binary entropy function
<Your code>
```

That is, the input is a matrix P, where the size determines the interpretation[1]. The basic idea is that each column in P corresponds to a probability distribution. If the column contains only one value, the script should derive the binary entropy function for this value. In this way the entropy for several distributions can be derived in a single call.

You can test your function with e.g.

```
>> Entropy([0.3 0.4 0.8; 0.1 0.3 0; 0.3 0.2 0.2; 0.3 0.1 0])
ans =
    1.8955    1.8464    0.7219
>> Entropy(1/8.*ones(8,1))
ans =
     3
>> plot(Entropy([0:0.1:1]));
```

[1]The MATLAB command size can be used to find the size of a matrix. The binary logarithm, $\log_2$ is called log2 in MATLAB.

where the last command should plot the binary entropy function. Compare your plot with e.g. the text book Figure 3.1 or the slides. Note that you need to take care of the case $0 \log 0 = 0$ separately. (Details on error message handling is optional.)

Mutual Information

The second function to be implemented is the mutual information. The header of the function is specified as

MutualInformation.m (File header)

```
function I=MutualInformation(P)
% The mutual information I(X;Y)
%
% P=P(X,Y) is the joint probability of X and Y.
<Your code>
```

The input matrix P is the joint probability distribution, $p(x, y)$. The MATLAB function `reshape` might come in handy, but it can be solved in other ways also.

Test your function with distributions from the course, e.g.

```
>> MutualInformation([0 3/4; 1/8 1/8]) % Example 3.9
ans =
    0.2936
>> MutualInformation([1/12 1/6 1/3; 1/4 0 1/6]) % Problem 3.5
ans =
    0.2503
```

The second example tests the case when P is not square, which should also work.

Hand in details

To hand in the problem send the m-files to `muhammad.umar_farooq@eit.lth.se` **and** `stefan.host@eit.lth.se`. Do not forget to write your name and STIL or student ID. Send the files as pure m-files, with no compression or any other fancy stuff. Your files will be tested according to the specifications and reply with the result pass or fail.