MASTER IN QUANTUM SCIENCE AND TECHNOLOGY

Quantum Information Theory
Homework Lecture 6

Francesc Sabater Marc Farreras Marcos Quintas Lluís Casabona

Consider the transmission of a classical random variable X through a classical-quantum channel with pure outputs such that the joint density matrix at the output of the channel is given by:

$$\boldsymbol{\rho_{XB} = \sum_x p_X(x)|x\rangle\langle x| \otimes |\theta_x\rangle\langle\theta_x|_B}$$

A measurement POVM is applied to the B share to yield Y. For the binary, uniformly distributed case the optimal POVM (in terms of minimizing the error probability) is obtained in the Lecture 5 slides and is given by $\{|+\rangle\langle+|, |-\rangle\langle-|\}$. In this case, the probability of error is $P_e = \frac{1}{2}(1-\sin\theta)$.

- Obtain and plot the accessible information $I(X;Y)$ and the quantum mutual information $I(X;B)_\rho$ for $\theta \in (0,\pi]$.

Let's first consider the following joint density matrix given in the exercise:

$$\rho_{XB} = \sum p_X(x)|x\rangle\langle x| \otimes |\theta_x\rangle\langle\theta_x|_B$$

If a measurement POVM is applied to the B share to yield Y we can write the corresponding resulting density matrix as:

$$\rho_{XY} = \sum_{x,y} p_X(x)|x\rangle\langle x| \otimes \mathrm{Tr}\{\wedge_y |\theta_x\rangle\langle\theta_x|\}|y\rangle\langle y| = \sum_{x,y} p_X(x)|x\rangle\langle x| \otimes p_{Y|X}(y|x)|y\rangle\langle y|$$

In the exercise we are told that we consider the binary uniformly distributed case, where $x \sim$ Bern $\left(\frac{1}{2}\right)$, $|\mathcal{Y}| = |\mathcal{X}| = 2$, and:

$$|\theta_0\rangle = \begin{pmatrix} \cos\theta/2 \\ \sin\theta/2 \end{pmatrix}, \quad |\theta_1\rangle = \begin{pmatrix} \cos\theta/2 \\ -\sin\theta/2 \end{pmatrix} \tag{1}$$

The optimal POVM (in terms of minimizing the error probability) for this case (and the one we will be using in this exercise) is given by $\{|+\rangle\langle+|, |-\rangle\langle-|\}$, where:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

while the probability of error is given by:

$$P_e = \frac{1}{2}(1 - \sin\theta)$$

We are asked to obtain and plot the accessible information $I(X;Y)$ and the quantum mutual information $I(X;B)_\rho$ for different values of θ. Let's start by computing the mutual information between X and B:

$$I(X;B)_\rho = H(B) - H(B|X) = H(B)$$

where it has been used that $H(B|X) = 0$ since given X, B is fully determined by state $|\theta_x\rangle$. The reduced density matrix of system B can be obtained tracing out X:

$$\rho_B = \mathrm{Tr}_x \left(\sum p_X(x) |x\rangle\langle x| \otimes |\theta_x\rangle\langle\theta_x|_B \right) = \sum_x p_X(x) |\theta_x\rangle\langle\theta_x|_B$$

Thus obtaining, for $x \sim \mathrm{Bern}\left(\frac{1}{2}\right)$ (where $p(x=0) = p(x=1) = 1/2$):

$$\rho_B = \frac{|\theta_0\rangle\langle\theta_0|_B + |\theta_1\rangle\langle\theta_1|_B}{2}$$

which can be re-expressed in matrix form using expressions (1), thus obtaining:

$$\rho_B = \begin{pmatrix} \cos^2(\theta/2) & 0 \\ 0 & \sin^2(\theta/2) \end{pmatrix}$$

This reduced density matrix of system B is already diagonal. As a result, one can directly write the entropy of B as:

$$H(B) = -p \log_2 p - (1-p) \log_2(1-p)$$

where in this case p can be defined as $p = \cos^2(\frac{\theta}{2})$, thus leading to:

$$H(B) = -\cos^2\left(\frac{\theta}{2}\right) \log \cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right) \log \sin^2\left(\frac{\theta}{2}\right)$$

Therefore, the quantum mutual information between X and B for state ρ_{XB} can be expressed as:

$$\boxed{I(B;X)_\rho = H(B) = -p \log_2 p - (1-p) \log_2(1-p), \quad \text{where } \; p = \cos^2\left(\frac{\theta}{2}\right)} \tag{2}$$

We now proceed to compute the mutual information $I(X;Y)$, which is defined as:

$$I(X;Y) = H(Y) - H(Y|X)$$

Let's remember again the resulting state after applying a POVM to system B to yield Y:

$$\rho_{XY} = \sum_{x,y} p_X(x) |x\rangle \langle x| \otimes \mathrm{Tr}\left\{\Lambda_y |\theta_x\rangle\langle\theta_x|\right\} |y\rangle \langle y| = \sum_{x,y} p_X(x) p_{Y|X}(y|x) |x\rangle\langle x| \otimes |y\rangle\langle y|$$

If we now compute the density matrix of system Y, we get:

$$\rho_y = \mathrm{Tr}_x\left(\rho_{xy}\right) = \sum_{x,y} p_x(x)\, \mathrm{Tr}\left(\Lambda_y |\theta_x\rangle\langle\theta_x|\right) |y\rangle\langle y|$$

where the two possible measurements $\{\Lambda_y\}$ correspond to the operators:

$$\Lambda_+ = |+\rangle\langle+| = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\Lambda_- = |-\rangle\langle-| = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Thus, we have:

$$p(y=0|x=0) = \mathrm{Tr}\left(\Lambda_+|\theta_0\rangle\langle\theta_0|\right) = \mathrm{Tr}\left(\frac{1}{2}\begin{bmatrix} \cos^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & \sin^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) \\ \cos^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & \sin^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\end{bmatrix}\right)$$

$$\Rightarrow p(y=0|x=0) = \frac{1}{2}(1+\sin\theta)$$

$$p(y=0|x=1) = \mathrm{Tr}\left(\Lambda_+|\theta_1\rangle\langle\theta_1|\right) = \mathrm{Tr}\left(\frac{1}{2}\begin{bmatrix} \cos^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & \sin^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) \\ \cos^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & \sin^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\end{bmatrix}\right)$$

$$\Rightarrow p(y=1|x=0) = \frac{1}{2}(1-\sin\theta)$$

$$p(y=1|x=0) = \mathrm{Tr}\left(\Lambda_-|\theta_0\rangle\langle\theta_0|\right) = \mathrm{Tr}\left(\frac{1}{2}\begin{bmatrix} \cos^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & -\sin^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) \\ -\cos^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & \sin^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\end{bmatrix}\right)$$

$$\Rightarrow p(y=1|x=0) = \frac{1}{2}(1-\sin\theta)$$

$$p(y=1|x=1) = \mathrm{Tr}\left(\Lambda_-|\theta_1\rangle\langle\theta_1|\right) = \mathrm{Tr}\left(\frac{1}{2}\begin{bmatrix} \cos^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & -\sin^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) \\ -\cos^2(\frac{\theta}{2}) - \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right) & \sin^2(\frac{\theta}{2}) + \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\end{bmatrix}\right)$$

$$\Rightarrow p(y=1|x=1) = \frac{1}{2}(1+\sin\theta)$$

where it has been used that $\sin(\theta) = 2\sin(\theta/2)\cos(\theta/2)$. At this point, where we know the conditional probabilities given y and x, we can treat the problem classically. We can also compute the total probabilities of obtaining a given y:

$$p(y=0) = p(y=0|x=0)p(x=0) + p(y=0|x=1)p(x=1) = \frac{1}{4}(1-\sin\theta+1+\sin\theta) = \frac{1}{2}$$

$$p(y=1) = p(y=1|x=0)p(x=0) + p(y=1|x=1)p(x=1) = \frac{1}{4}(1+\sin\theta+1-\sin\theta) = \frac{1}{2}$$

Thus, Y is a also a Bernoulli one-half and thus we have:

$$H(Y) = 1$$

Finally, we can compute $H(Y|X)$ as:

$$H(Y|X) = p(x=0)H(Y|x=0) + p(x=1)H(Y|x=1) = \frac{1}{2}(H(Y|x=0) + H(Y|x=1))$$

where:

$$H(Y|x=0) = -\frac{1-\sin\theta}{2}\log\left(\frac{1-\sin\theta}{2}\right) - \frac{1+\sin\theta}{2}\log\left(\frac{1+\sin\theta}{2}\right)$$

$$H(Y|x=1) = -\frac{1-\sin\theta}{2}\log\left(\frac{1-\sin\theta}{2}\right) - \frac{1+\sin\theta}{2}\log\left(\frac{1+\sin\theta}{2}\right)$$

And as a result:

$$H(Y|X) = -\frac{1-\sin\theta}{2}\log\left(\frac{1-\sin\theta}{2}\right) - \frac{1+\sin\theta}{2}\log\left(\frac{1+\sin\theta}{2}\right)$$

Remembering that $P_e = \frac{1-\sin\theta}{2}$, we can write:

$$H(Y|X) = -P_e \log(P_e) - (1-P_e)\log(1-P_e) \equiv H(P_e)$$

Therefore, taking into account the previous results, the mutual information between X and Y can be expressed as:

$$\boxed{I(Y;X) = 1 - H(P_e)\,, \quad \text{with } P_e = \frac{1-\sin\theta}{2}} \tag{3}$$

From these results, one can plot the mutual information $I(X;B)_\rho$ presented in Eq. (2) and the mutual information $I(X;Y)$ presented in Eq. (3) as a function of the parameter θ. The results obtained in this case can be found in Figure 1.

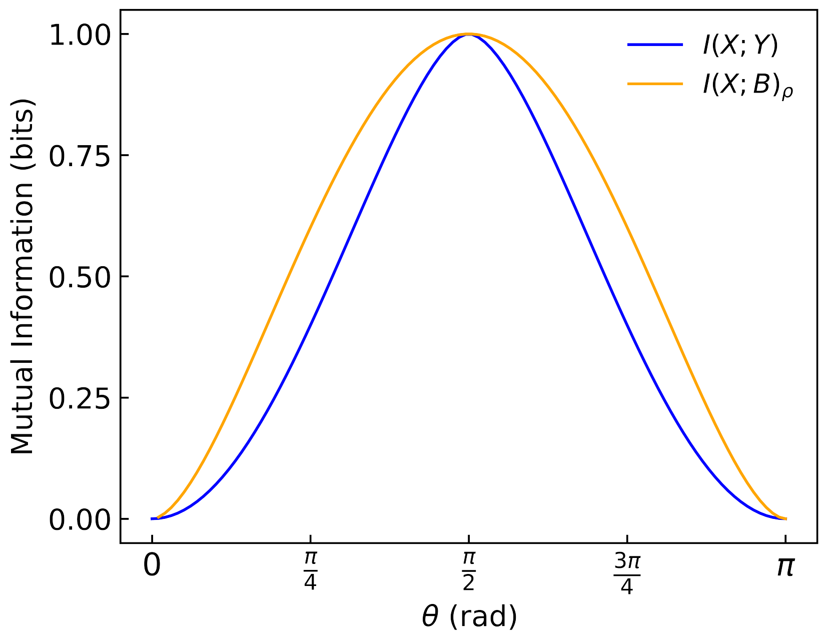

Figure 1: Accessible mutual information $I(X;Y)$, and quantum mutual information $I(X;B)_\rho$ as a function of parameter θ.

Let's now briefly comment some observations on the results obtained. From Fig. (1) it is possible to see that $I(X;B) \geq I(X;Y)$. The non-trivial equality is fulfilled at $\theta = \pi/2$. The fact that the mutual information between X and B is always larger or equal than the one between X and Y is due the the quantum data processing inequality. Since we are acting on system B with a POVM to obtain Y we will always have less or equal information in the X, Y system compared to the X, B system.

We now consider $X \sim$ Unif([0,1,2,3]), $|\mathcal{Y}| = |\mathcal{X}| = 4$. For every random variable realization x we use three parallel quantum channels like the one employed before such that:

$$\boldsymbol{\rho_{XB} = \sum_x p_X(x)|x\rangle \langle x| \otimes |\psi_x\rangle \langle \psi_x|_B}$$

where:

$$\boldsymbol{\begin{aligned}|\psi_0\rangle_{B^3} &= |\theta_0\rangle_B \otimes |\theta_0\rangle_B \otimes |\theta_0\rangle_B \\ |\psi_1\rangle_{B^3} &= |\theta_0\rangle_B \otimes |\theta_1\rangle_B \otimes |\theta_1\rangle_B \\ |\psi_2\rangle_{B^3} &= |\theta_1\rangle_B \otimes |\theta_0\rangle_B \otimes |\theta_1\rangle_B \\ |\psi_3\rangle_{B^3} &= |\theta_1\rangle_B \otimes |\theta_1\rangle_B \otimes |\theta_0\rangle_B .\end{aligned}}$$

Again, a measurement POVM is applied to the B^3 share to yield Y:

$$\rho_{XY} = \sum_{x,y} p_X(x)|x\rangle \langle x|_X \otimes \operatorname{tr}\{\Lambda_y|\psi_x\rangle\langle\psi_x|_{B^3}\}|y\rangle \langle y|_Y$$

In this case, the POVM known as the square-root measurement becomes optimal meaning:

$$\Lambda_y = \frac{1}{4}\left(\rho_{B^3}\right)^{-\frac{1}{2}} |\psi_y\rangle \langle\psi_y| \left(\rho_{B^3}\right)^{-\frac{1}{2}}, \text{for} y \in [0,1,2,3]$$

and where $\rho_{\mathrm{B^3}} = \mathbf{tr_X}\{\rho_{\mathrm{XB^3}}\}$. Note that since ρ_{B^3} is not full rank the inverse operation should be replaced by the pseudo inverse, available in any numerical software.

- Show that $\{\Lambda_y\}$ is a proper POVM.

- Show that $\{\Lambda_y\}$ is a proper POVM.

- Obtain (numerically) and plot the accessible information $I_3(X;Y)$ and the quantum mutual information $I_3\left(X;B^3\right)_\rho$ for $\theta \in (0,\pi]$.

- Finally plot $I_3(X;Y) - 3I(X;Y)$ and $I_3\left(X;B^3\right)_\rho - 3I(X;B)_\rho$ for $\theta \in (0,\pi]$.

First of all we need to prove that the given operators $\{\Lambda_y\}$ are a proper POVM. For this, we have to check that the operators are positive, hermitian, and that fulfill the completeness relation. Let's start with checking if the operators are hermitian. If one calculates the corresponding dagger operators, it is possible to find that:

$$\Lambda_y^\dagger = \frac{1}{4}\left(\left(\rho_{B^3}\right)^{-\frac{1}{2}} |\psi_y\rangle \langle\psi_y| \left(\rho_{B^3}\right)^{-\frac{1}{2}}\right)^\dagger = \frac{1}{4}\left(\left(\rho_{B^3}\right)^{-\frac{1}{2}}\right)^\dagger |\psi_y\rangle \langle\psi_y| \left(\left(\rho_{B^3}\right)^{-\frac{1}{2}}\right)^\dagger$$
$$= \frac{1}{4}\left(\rho_{B^3}\right)^{-\frac{1}{2}} |\psi_y\rangle \langle\psi_y| \left(\rho_{B^3}\right)^{-\frac{1}{2}} = \Lambda_y$$

where it has been used that $\rho_{B^3} = \sum_x P_X(x)|\psi_x\rangle \langle\psi_x | = \frac{1}{4}\sum_x | \psi_x\rangle \langle\psi_x|$ is an hermitian matrix, and thus the square root of its inverse matrix is also hermitian. Thus we have that:

$$\Lambda_y^\dagger = \Lambda_y \qquad \text{for } y \in [0,1,2,3]$$

which means that the operators in the set $\{\Lambda_y\}$ are indeed hermitian.

It is also possible to check that the sum of the operators in the set gives the identity, as:

$$\sum_y \Lambda_y = \sum_y \frac{1}{4} (\rho_{B^3})^{-\frac{1}{2}} |\psi_y\rangle \langle\psi_y| (\rho_{B^3})^{-\frac{1}{2}} = (\rho_{B^3})^{-\frac{1}{2}} \left(\frac{1}{4}\sum_y |\psi_y\rangle \langle\psi_y|\right) (\rho_{B^3})^{-\frac{1}{2}} =$$
$$= (\rho_{B^3})^{-\frac{1}{2}} \rho_{B^3} (\rho_{B^3})^{-\frac{1}{2}} = \mathbb{1}$$

where the definition of ρ_{B^3} has been used (the term $1/4$ comes from the uniform probability of the four possible values of x). The fact that $\{\Lambda_y\}$ are positive definite operators can also be shown by computing:

$$\langle\Phi|\Lambda_y|\Phi\rangle = \frac{1}{4}\langle\Phi| (\rho_{B^3})^{-\frac{1}{2}} |\psi_y\rangle\langle\psi_y| (\rho_{B^3})^{-\frac{1}{2}} |\Phi\rangle = \frac{1}{4}||\langle\Phi| (\rho_{B^3})^{-\frac{1}{2}} |\psi_y\rangle||^2 \geq 0$$

where it has been used that the norm of something can not be negative. With all these results, we have shown that $\{\Lambda_y\}$ is a proper POVM.

Let's now proceed to obtain numerically the accessible information $I_3(X;Y)$ and the quantum mutual information $I_3(X;B^3)_\rho$. Let's start with $I_3(X;B^3)_\rho$. We know that ρ_{XB^3} is a quantum-classical state, and therefore the mutual information $I_3(X;B^3)$ can be expressed as:

$$I_3\left(X;B^3\right) = H\left(\rho_{B^3}\right) - \sum_x P_X(x) \underbrace{H\left(\rho^x_{B^3}\right)}_{0} = H\left(\rho_{B^3}\right) = -\operatorname{tr}\left\{\rho_{B^3}\log\left(\rho_{B^3}\right)\right\} \tag{4}$$

where it is used that $H\left(\rho^x_{B^3}\right) = 0$ because $\rho^x_{B^3}$ is pure for all possible values of x. Therefore, we have to compute, for each angle, the density matrix $\rho_{B^3} = \frac{1}{4}\sum_y |\psi_y\rangle\langle\psi_y|$, its eigenvalues, and apply expression (4).

On the other hand, for the mutual information $I_3(X;Y)$, one can apply the following definition of mutual information:

$$I_3(X;Y) = H(Y) - H(Y|X)$$

where the conditional entropy is defined as:

$$H(Y|X) = \sum_x p_X(x) \cdot H(Y|X=x) = -\sum_x p_X(x) \sum_y p_{Y|X}(y|x)\log(p(y|x))$$

and:

$$H(Y) = -\operatorname{tr}\left\{\rho_Y \log\left(\rho_Y\right)\right\}; \quad \rho_Y = \operatorname{tr}_X\left\{\rho_{XY}\right\} = \frac{1}{4}\sum_{x,y} p_{Y|X}(y|x)|y\rangle\langle y|$$

where in both cases the term $p_{Y|X}(y|x)$ can be computed as:

$$p_{Y|X}(y|x) = \operatorname{tr}\left\{\Lambda_y |\psi_x\rangle\langle\psi_x|\right\}.$$

After this explanation on how to obtain the desired quantities, we proceed to show the numerical results obtained using the programming language *Python*. First, a comparison between $I_3(X;B^3)_\rho$ and $I_3(X;Y)$ as a function of the value of θ can be seen in Figure 2.

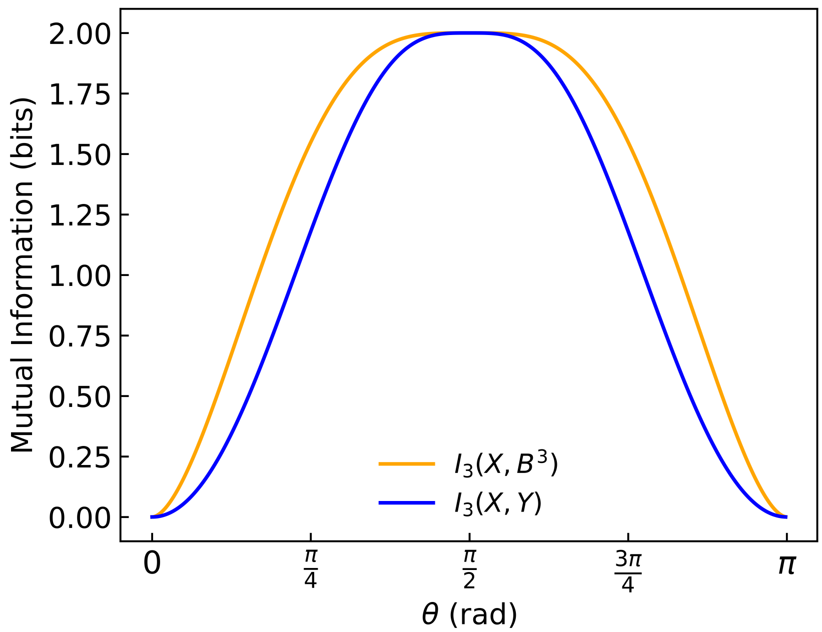

Figure 2: $I_3(X;B^3)_\rho$ and $I_3(X;Y)$ as a function of θ.

From Figure 2 it can be observed that now the maximal value for the mutual information is achieved for a slightly wider region of angles centered around $\theta = \pi/2$ when compared to the previous case (corresponding to Figure 1) and that the data processing inequality is still fulfilled, as for all θ values:

$$I_3(X;B^3)_\rho \geq I_3(X;Y)$$

as was to be expected. Moreover, a comparison has been conducted between the mutual information of the triple channel and that of the three parallel channels, by plotting $I_3(X;Y) - 3I(X;Y)$ and $I_3(X;B^3)_\rho - 3I(X;B)_\rho$ for different values of θ. The results obtained can be found in Figure 3.

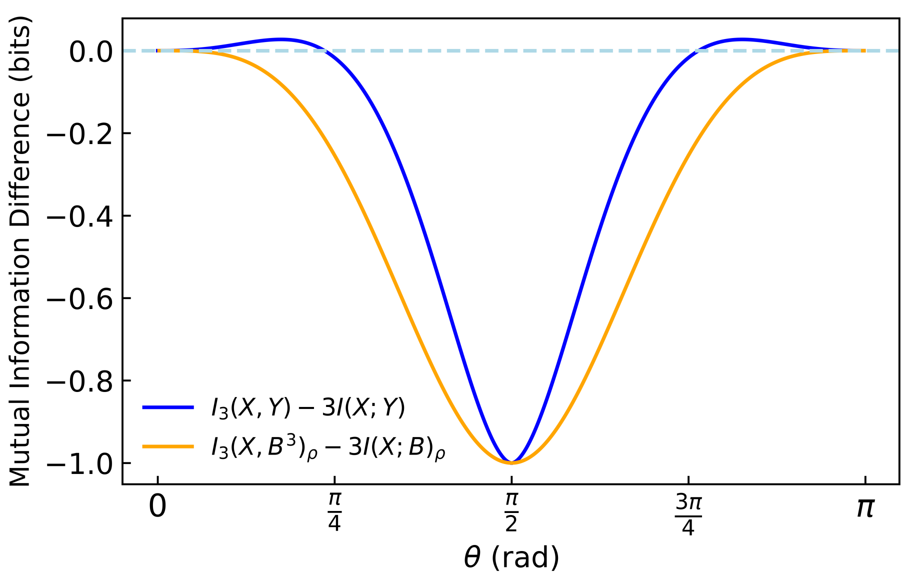

Figure 3: Comparison between the use of three parallel channels and using a single channel thrice.

The results in Figure 3 show that for the classical-quantum variables X and B, the difference $I_3(X; B^3)_\rho - 3I(X; B)_\rho$ is negative or equal to zero for all possible values of θ. This fact implies that there is more information in 3 times the two-dimensional system than in 1 time the four-dimensional system. However, it can also be observed that if one performs a measurement transforming the original classical-quantum system (given by ρ_{XB} or ρ_{XB^3} to another system associated to classical variables X, Y (ρ_{XY}) the sign of the function $I_3(X; Y) - 3I(X; Y)$ depends on the angle θ. For θ values close to 0 or π the function is positive, thus indicating that one has more information by using the four-dimensional system instead of three times the two-dimensional one.

On the other hand, for θ values around $\pi/2$ the function is negative (same behaviour as with the mutual information between X and B previously tackled). If one further analyzes the results, it is seen that both functions present a minimum value at $\theta = \pi/2$ (angle at which the maximum mutual information is reached, corresponding to a binary symmetric channel with uniform probabilities). For the single channel the maximum is 1 bit (as seen in Figure 1), while for the three parallel channels the maximum value is 2 bits (seen in Figure 2). As a result, in the region around $\theta = \pi/2$ it is preferable to use just one channel as opposed to the regions near the sides of the plot, as previously tackled.

Codes

Exercise 1

```
# EXERCISE 1 (Figure 1)
# I(X;Y) and I(X;B) versus theta

import numpy as np
import matplotlib.pyplot as plt

def pe(theta):
        return (1/2)*(1-np.sin(theta))
def H(p):
        return -p*np.log2(p)-(1-p)*np.log2(1-p)
def p2(theta):
        return np.cos(theta/2)**2

theta=np.linspace(0.00,np.pi,num=480)

xy=[1-H(pe(k)) for k in theta]
xb=[H(p2(k)) for k in theta]

# Let's generate the plot of Figure 1
theta_ticks = [0, np.pi/4, np.pi/2, 3*np.pi/4, np.pi]
theta_labels = ['0', r'$\frac{\pi}{4}$', r'$\frac{\pi}{2}$', r'$\frac{3\pi}{4}$', r'$\pi$']
plt.figure(figsize=(5.2,4))
plt.plot(theta,xy,linewidth=1.2, label=r'$I(X;Y)$', color="b")
plt.plot(theta,xb,linewidth=1.2, label=r'$I(X;B)_\rho$', color="orange")
plt.legend(loc='best',fontsize = 11,frameon=False)
plt.tick_params(axis ="x", direction = "in")
plt.tick_params(axis ="y", direction = "in")
plt.xlabel(r'$\theta$ (rad)',fontsize = 12)
plt.ylabel("Mutual Information (bits)",fontsize = 12)
plt.xticks(theta_ticks, labels=theta_labels, fontsize = 13)
plt.yticks(fontsize = 12)
plt.savefig('fig1.png', format='png', bbox_inches = "tight", transparent=True, dpi=1200)
plt.show()
```

Exercise 2

```
# EXERCICI 2

# 2.1. I3(X;Y) and I3(X;B3) versus angle theta

import numpy as np
from scipy import linalg
from scipy.linalg import sqrtm
import math
import matplotlib.pyplot as plt

pi= np.pi

# Creem els dos vectors theta0 i theta1
#Function that creates the ket theta0 as a 1D array
def ket_theta0(theta):
    alfa=theta/2.
    vector0=np.array([np.cos(alfa),np.sin(alfa)])
    return vector0

#Function that creates the ket theta1 as a 1D array
```

```
def ket_theta1(theta):
    alfa=theta/2.
    vector1=np.array([np.cos(alfa),-np.sin(alfa)])
    return vector1

#_____________________________________________________________________________

#Functions which creates the kets psi_i where i=0,1,2,3 as a 1D array

def ket_psi0(theta):
    vect0=ket_theta0(theta)
    vect1=np.copy(vect0)
    vect2=np.copy(vect0)
    vect_inter=np.kron(vect1,vect2)
    vect_fin=np.kron(vect0,vect_inter)
    return vect_fin


def ket_psi1(theta):
    vect0=ket_theta0(theta)
    vect1=ket_theta1(theta)
    vect2=np.copy(vect1)
    vect_inter=np.kron(vect1,vect2)
    vect_fin=np.kron(vect0,vect_inter)
    return vect_fin


def ket_psi2(theta):
    vect0=ket_theta1(theta)
    vect1=ket_theta0(theta)
    vect2=np.copy(vect0)
    vect_inter=np.kron(vect1,vect2)
    vect_fin=np.kron(vect0,vect_inter)
    return vect_fin

def ket_psi3(theta):
    vect0=ket_theta1(theta)
    vect1=np.copy(vect0)
    vect2=ket_theta0(theta)
    vect_inter=np.kron(vect1,vect2)
    vect_fin=np.kron(vect0,vect_inter)
    return vect_fin

#_____________________________________________________________________________
#Create the density matrix of the pure state

def density_matrix(vector):
    matrix_pure=np.outer(vector,vector)
    return matrix_pure

#_____________________________________________________________________________

#Compute the matrix density_B3
def density_B3(theta):
    psi0=ket_psi0(theta)
    density_ket0=density_matrix(psi0)
    psi1=ket_psi1(theta)
    density_ket1=density_matrix(psi1)
    psi2=ket_psi2(theta)
    density_ket2=density_matrix(psi2)
```

```
    psi3=ket_psi3(theta)
    density_ket3=density_matrix(psi3)
    final_matrix=(1/4.)*(density_ket0+density_ket1+density_ket2+density_ket3)
    return final_matrix

#____________________________________________________________________________

#Function that computes the Von Neuman entropy of a given matrix
def Entropy(density_matrix):
    #eigenvalues_dens=np.diag(density_matrix)
    eigenvalues_dens=np.linalg.eigvals(density_matrix)
    entropy_value=0.
    for i in eigenvalues_dens:
        if i >= 1e-40:
            entropy_value+=-i*np.log2(i)
    return entropy_value
#____________________________________________________________________________

#____________________________________________________________________________
#Function that computes the POVMs:

def POVMs_Y(theta):
    matrix_B3=density_B3(theta)
    sqrt_matrix=sqrtm(matrix_B3)
    a =math.isnan(sqrt_matrix[0][0])
    #print(a)
    if a==True:
        list_POVM=0.
        list_pureMatrix=0.
    else:
    #identity0=0.00000000000000005*np.identity(8)
    #matrix_B3=matrix_B3+identity0
    #sqrt_B3=sqrtm(matrix_B3)
    #print(theta)
    #print(matrix_B3)
        #print('')
        #print('B3 matrix')
        #print(matrix_B3)
        sqrt_invB3=linalg.pinv(sqrt_matrix).real
        #inverse_B3= linalg.pinv(matrix_B3)

        #sqrt_invB3=sqrtm(inverse_B3)


        psi0=ket_psi0(theta)
        density_ket0=density_matrix(psi0)
        psi1=ket_psi1(theta)
        density_ket1=density_matrix(psi1)
        psi2=ket_psi2(theta)
        density_ket2=density_matrix(psi2)
        psi3=ket_psi3(theta)
        density_ket3=density_matrix(psi3)

        list_pureMatrix=[density_ket0,density_ket1,density_ket2,density_ket3]

        POVM00=np.matmul(density_ket0,sqrt_invB3)
        POVM0=(1/4.)*np.matmul(sqrt_invB3,POVM00)

        POVM11=np.matmul(density_ket1,sqrt_invB3)
        POVM1=(1/4.)*np.matmul(sqrt_invB3,POVM11)
```

```
        POVM22=np.matmul(density_ket2,sqrt_invB3)
        POVM2=(1/4.)*np.matmul(sqrt_invB3,POVM22)

        POVM33=np.matmul(density_ket3,sqrt_invB3)
        POVM3=(1/4.)*np.matmul(sqrt_invB3,POVM33)

        list_POVM=[POVM0,POVM1,POVM2,POVM3]

    return list_POVM, list_pureMatrix

#_____________________________________________________________________________

#Function that computes the conditional probability p(y,x)

def conditional_XY(list_POVM_Y,list_density_X,x,y):

        POVM_Y=list_POVM_Y[y]
        density_X = list_density_X[x]
        matrix=np.matmul(POVM_Y,density_X)
        pcond_XY=np.trace(matrix)
        return pcond_XY

#______________________________________________________________________________

#Function that computes the entropy of H(Y):

def Entropy_Y(list_POVM_Y,list_density_X):
    entropy=0.
    py=0.
    for y in range(0,4):
        for x in range(0,4):
            py+=(1/4.)*conditional_XY(list_POVM_Y, list_density_X, x, y)
        if py >= 1e-36:
            entropy+=-py*np.log2(py)
        py=0.

    return entropy

#______________________________________________________________________________

def Entropy_cond_XY(list_POVM_Y,list_density_X):
    entropy=0.
    py=0.
    for y in range(0,4):
        for x in range(0,4):
            py= conditional_XY(list_POVM_Y, list_density_X, x, y)
            if py > 1e-10:
               entropy+=-(1/4.)*py*np.log2(py)
        py=0.
    return entropy


list_infXB3=[]
list_teta=[]

delta_teta=pi/500.

#------------------------------------------------------------------------------
#Compute I(X,B3) information
```

```python
for m in range(0,500):
    teta=m*delta_teta
    list_teta.append(teta)
    matrix_B3=density_B3(teta)
    information_XB3=Entropy(matrix_B3)
    list_infXB3.append(information_XB3)


#-----------------------------------------------------------------------------
#Compute I(X,Y) information

list_infXY=[]
list_teta2=[]
list_HY=[]
list_HYX=[]

for m in list_teta:
    teta=m

    #Compute the POVM list and psi_x density matrix:
    list_POVM,list_psix =POVMs_Y(teta)
    if list_POVM==0.:
        #print('skip teta')
        continue

    #Compute H(y):
    #print('not skip teta')
    H_Y=Entropy_Y(list_POVM,list_psix)

    #Compute H(Y|X):
    H_YX=Entropy_cond_XY(list_POVM,list_psix)

    #Compute mutual information:
    I_XY=H_Y-H_YX
    list_teta2.append(teta)
    list_infXY.append(I_XY)
    list_HY.append(H_Y)
    list_HYX.append(H_YX)

theta_ticks = [0, np.pi/4, np.pi/2, 3*np.pi/4, np.pi]
theta_labels = ['0', r'$\frac{\pi}{4}$', r'$\frac{\pi}{2}$', r'$\frac{3\pi}{4}$', r'$\pi$']
plt.figure(figsize=(5.2,4))
plt.ylabel("Mutual Information (bits)",fontsize = 12)
plt.xlabel(r'$\theta$ (rad)',fontsize = 12)
plt.plot(list_teta,list_infXB3,'orange',label=r"$ I_{3}(X,B^{3})$")
plt.plot(list_teta2,list_infXY,'b',label=r"$ I_{3}(X,Y)$")
plt.tick_params(axis ="x", direction = "in")
plt.tick_params(axis ="y", direction = "in")
plt.legend(loc='best',fontsize = 11,frameon=False)
plt.xticks(theta_ticks, labels=theta_labels, fontsize = 13)
plt.yticks(fontsize = 12)
plt.savefig('fig2.png', format='png', bbox_inches = "tight", dpi=1200)
plt.show()
```

```python
# EXERCISE 2
# 2.2. Plot I3(X;Y) - 3I(X;Y) and I3(X;B) - 3I(X;B)

N = 480

#print(list_teta2)
def pe(theta):
        return (1/2)*(1-np.sin(theta))
def H(p):
        return -p*np.log2(p)-(1-p)*np.log2(1-p)
def p2(theta):
        return np.cos(theta/2)**2
theta=list_teta2
theta2=list_teta
xy=[1-H(pe(k)) for k in theta]
xb=[H(p2(k)) for k in theta2]

difB = list_infXB3 - 3*np.array(xb)
difI = list_infXY -3*np.array(xy)

# Let's plot the results
theta_ticks = [0, np.pi/4, np.pi/2, 3*np.pi/4, np.pi]
theta_labels = ['0', r'$\frac{\pi}{4}$', r'$\frac{\pi}{2}$', r'$\frac{3\pi}{4}$', r'$\pi$']
plt.figure(figsize=(5.8,3.5))
plt.ylabel("Mutual Information Difference (bits)",fontsize = 11)
plt.xlabel(r'$\theta$ (rad)',fontsize = 12)
plt.plot(list_teta2,difI,'blue',label=r"$ I_{3}(X,Y)-3I(X;Y)$")
plt.plot(theta2,difB,'orange',label=r"$ I_{3}(X,B^{3})_\rho - 3 I(X;B)_\rho$")
plt.tick_params(axis ="x", direction = "in")
plt.tick_params(axis ="y", direction = "in")
plt.legend(loc='best',fontsize = 10,frameon=False)
plt.xticks(theta_ticks, labels=theta_labels, fontsize = 13)
plt.axhline(y=0, color='lightblue', linestyle='--')
plt.yticks(fontsize = 12)
plt.savefig('fig2.png', format='png', bbox_inches = "tight", dpi=1200)
plt.show()
```