relative-entropy-expressions

February 15, 2022

0.1 Expressions for Relative Entropy and their (non) equivalence

There is a one-one correspondence between density matrices and just matrices. For example,

$$|0\rangle\langle 0| - \frac{1}{3}|0\rangle\langle 1| - \frac{1}{3}|1\rangle\langle 0| + |1\rangle\langle 1| \leftrightarrow \begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix}$$

so the entire analysis can be done with regular matrices.

Consider the two matrices ρ and σ being regular density matrices, this means that they are positive $\implies$ they are hermitian, thus diagonalizable, and have positive eigenvalues. Consider the expressions,

$$D_1 = tr(\rho \log{(\rho\sigma^{-1})}) D_2 = tr(\rho \log{(\sigma^{-1}\rho)})\text{and, } D_3 = tr(\rho \log{(\rho)} - \rho \log{(\sigma)}).$$

As, $\rho = \rho^\dagger$ and $\sigma = \sigma^\dagger$,

$$D_1^* = tr((\rho \log{(\rho\sigma^{-1})})^\dagger) = tr((\log{(\rho\sigma^{-1})}^\dagger \rho^\dagger)$$

as log has a taylor expansion, $\log{(A)}^\dagger = \log{(A^\dagger)}$, and also, use the cyclicity of trace,

$$= tr(\rho \log{(\sigma^{-1^\dagger}\rho^\dagger)}) = D_2$$

This means that D_1 and D_2 are complex conjugates of each other. However, D_3 is always real. Let,

$$\rho = \Sigma_j \rho_j |j\rangle\langle j|$$

be a diagonal expansion of ρ in orthonormal basis $\{|i\rangle\}$ and $\sigma = \Sigma_j \sigma_j |\tilde{j}\rangle\langle\tilde{j}|$ be a diagonal expansion of σ in orthonormal basis $\{|\tilde{i}\rangle\}$. Consider,

$$D_3 = tr(\rho\log\rho) - tr(\rho\log\sigma) = \Sigma_i \rho_i \log\rho_i - \Sigma_{ij}\rho_i \log\sigma_j |\langle i|\tilde{j}\rangle| \in \mathbb{R}$$

Now to see if D_1 can ever be complex and whether, when real, it can equal D_3. This is just done numerically.

[8]:
```
# Getting a density matrix. Using sage so it's easy to get matrices in differnt␣
↪rings.
def make_matrix(ring = CC, dim =2):
    A = random_matrix(ring, dim, dim) # a random 3x3 matrix with complex␣
↪coefficients
```

```
    A = A.conjugate_transpose()*A # to make it positive and hermitian
    return A/A.trace() # to normalize it
```

[16]:
```
A = make_matrix(CC, 2)
B = make_matrix(CC, 2)
```

[23]:
```
print(A.eigenvalues(), B.eigenvalues()) # just to check that the matrices are␣
 ↪properly defined
```

```
[0.769770705966512, 0.230229294033488] [0.833076945423677, 0.166923054576323]
```

[13]:
```
from scipy.linalg import logm # to take matrix logarithm
import numpy as np
```

[19]:
```
# matches expressions in the write up
d_1 = A*logm(A*B.inverse())
d_2 = A*logm(B.inverse()*A)

D_1 = d_1.trace()
D_2 = d_2.trace()
```

[20]:
```
print(D_1, D_2)
```

```
(0.29299315704878875+0.022180207594100495j)
(0.29299315704878826-0.02218020759410028j)
```

D_1 and D_2 are complex and are conjugates of each other! Now check D_3,

[21]:
```
d_3 = A*logm(A) - A*logm(B)

D_3 = d_3.trace()
```

[22]:
```
print(D_3)
```

```
(0.28407999822370644-3.8808249980510473e-17j)
```

It is real! (up to numerical accuracy of course). It is of the same order of magnitue as D_1 but not the same.

The results are slightly different for real matrices, and even more interesting in my opinion!

[26]:
```
C = make_matrix(QQ, 3) # choosing rational matrices for exact results
D = make_matrix(QQ, 3)
```

[27]:
```
print(C.eigenvalues(), D.eigenvalues())
```

```
[20/49, 0.06300587325483017?, 0.5288308614390475?] [0.05160300406552340?,
0.2964632548685508?, 0.6519337410659259?]
```

[30]:
```
d_1 = C*logm(C*D.inverse())
d_2 = C*logm(D.inverse()*C)
```

```
D_1 = d_1.trace()
D_2 = d_2.trace()
```

[31]:
```
print(D_1, D_2)
```

```
0.24912140097453755 0.24912140097453825
```

They equal each other of course being real and complex conjugates of each other. What about D_3?

[32]:
```
d_3 = C*logm(C) - C*logm(D)

D_3 = d_3.trace()
```

[33]:
```
print(D_3)
```

```
0.1885573555351014
```

It's different from D_1!

The above was for real valued complex matrices, that is, matrices that belong in $\mathbb{C}^d$ but just have real values. Dealing with $\mathbb{R}^d$ is a completely different thing because hermiticity and diagonalizability are not guaranteed there.

Of course, if the matrices are simultaneosly diagonalizable, they would have the same vales of D_1, D_2 and D_3, which is what the classical probabilty densities correspond to. There is the unique canonical basis in which the probability densities are diagonal.