CS 6817 HW 2

2.38, 2.46, 2.56 (a) and (b)

# 2.38

We use this formula of stability:

The formula for Fourier coefficients of the Tribes function is given in the textbook on page 97:

There are tribes with members each. For each , we write it as , where is the intersection of with the i-th tribe:

Since the we need in the stability formula:

If we define , then the above could be rewritten as

Since when . Notice both instances have in common.

So,

We want to simplify something:

So,

The Piazza hint says to take asymptotics into account. I’m not sure, but so maybe

# 2.46

The Mean Value Theorem for a function continuous in the domain interval is:

We propose that

So, for some :

Which means we have to show

We differentiate with respect to the noise parameter :

Because when , the expression inside the summation is anyway.

Now we set and employ knowledge from the previous exercise, 2.45:

Which makes

Now we need to prove

Well, since and ,

Thus,

QED.

# 2.56

## (a)

Ideally, we want , which means ideally, . So, we want as high as possible.

Where is the parity of for the -th party.

If and , then . And because of the Cauchy-Schwarz inequality,

Because

So

Which means

Because of Parseval’s identity, which makes this sum a probability distribution on .

Thus, we want to maximize

Which means maximizing

Because , , is maximized when is as small as possible. And since we can’t have a trivial solution, has to be greater than , so , **so and must be dictator functions**:

or for some

and

or for some

And we need

To be as large as possible

We said earlier that

Needs to be greater than . To make this as likely as possible, we need the two dictator functions to be dictated by the same bit because we have no guarantees about choosing two different bits and since they’re independent of each other; and we need them to be either both positive dictators or negative dictators because , whereas , and because and are actually more likely to be equal than unequal:

Because they’re equal when they both get right or both get wrong

Because they’re unequal when one of them gets right and the other gets it wrong

And the functions themselves need to be:

and for some

or

and for some

i.e. dictated by the same bit, and both multiplied by or .