14-5

One zero something so you say it would never be that y1 is one and y2 is zero

y1 is one means that one of this guys is gonna be owned

so still it could be y2 is zero

but you are getting close

ok 1000 whatever this will be can not happen

Why? Because if there is one here is either x1 or x2 are gonna be owned

And that means that it cannot be that all these three would be zero

At least one of them must be owned

Right? So an example that start with 1000 would never happen

So I should be able to tolerate different probability distribution

Otherwise, there is a lot of things that I cannot do algorithmically

So this is an important point that gives rise to y PAC-Learning the strict definition

But why it's important that it is striating this way

We are going to see at least one more example of this but this is an important issue

OK! So now we have this we can talk about a lot of other examples. Right?

So we prove so far conjunctions, K-CNF

Here is another interesting example

So if I have an unbiased learning that is my hypothesis space is all the boolean functions on n features, how many boolean functions on n features are total?

two to the two to the n (2^(2^n))

Right? There are 2^n instances and therefore the number of function on 2^n instances is (2^(2^n)) right? we have seen this before

That means the log of this is 2^n so the bound for m is exponential which is not PAC-Learnable.

Right? Information theoretically it is not PAC-Learnable

Ah. Now we can think about many other functions

K-CNF we just talked about and define many other classes of functions which I am defining here and we are not gonna derive all of them

Ah in each case you can think about this class of functions

Ah count how many elements are there and see that this gonna be PAC-Learnable

Ah and I am giving this counts and I don't wanna stop on this

I wanna talk about one class of functions just because there is another interesting points to make

and this is going to be this class of functions K-term DNF

So if I call it DNF it means that it is a disjunction of conjunctions

So it's this this this this and this or this this this this and this or this this this this and this

Right? So it is a disjunction of conjunctions

But it is a disjunction up to k conjunctions

OK? So in fact I am gonna talk about the very simple case of a disjunction up to 2 conjunctions so it's just ah two-term DNF which means you have two conjunctions and an or between them very simple function

OK? And the question is can we PAC-Learn this?

And if you do the counting, Ah you would say yes

And I am gonna give you this as your homework

Ah because we have counted K-CNF and it's gonna be easy for you to count K-DNF and therefore also K-term DNF

But the point that I wanna make is that even though counting wise the class is small enough and if you will be able to develop an algorithm consistent with this think about the very simple class. Right? You have two disjunctions

14-6

Within all between them either this conjunction or this conjunction make the example positive

Think about if you want in the linear separation step right?

So think about you have two linear separator

Each one of them counts on with an error that says on which side is positive

Right you imagine what I am saying?

So here is the hypothesis space with the error this way which means here is positive other side is negative

take another one like this, linear functions, this side is positive this side is negative

Now take the union of this So an example of this is positive if it’s either the positive side of this line or that line

So this is kind of the consequence we are talking about right?

Just easier to do it in the finite combinatory space because I am goanna to prove something on this but at least argue something

Even though from sample complexity perspective things are good and I am letting you verify this it’s the same counting argument that I used before for K-CNF

Ah they are not good for the computational complexity in this cases

What do I mean? What I mean is that I am goanna give you data and tell you this data comes from a two-term DNF that it is consistent with a two-term DNF

There exist a two-term DNF that all the examples are positive in this data set are positive for this two-term DNF all those that are negative in this data set are negative for this two-term DNF

But finding this two-term DNF that is consistent with this data set is NP-hard

Right? So I am not goanna to prove this and this is by now a 30 years old theory and actually Lenin Pete’s proved when he was a post-doc

A very important results from this perspective because it shows that our cases where information theoretically it’s OK but computational complexity prevents us from learning it

Right? So the question is that what we do?

So it’s PAC-Learnable because of this part but its not efficiently PAC-Learnable due to computational complexity.

What we are goanna do here?

In fact, what we are goanna is that we are going to do the same trick we’ve done from conjunctions to linear separable. We are going to enlarge the hypothesis space.

So we talked about it last time which hypothesis space do we like?

Do we like the small or large?

And the tendency if you look at all incoming results, we like small hypothesis spaces.

Right? Because the impact m, the number of examples we need to use in order to learn

Nonetheless, it brings up some combotary.

So we are goanna say because K-CNF is a superset of K-DNF. That means

What does it mean? We talk about classes of functions so I want to make sure that we understand what it means. That means if you take a function that is represented as a K term DNF, you can also represent it as K term CNF and I am goanna show this in a second but because this is true because we know how to learn K-CNF we are done

You understand this structure of argument? Any questions on this?