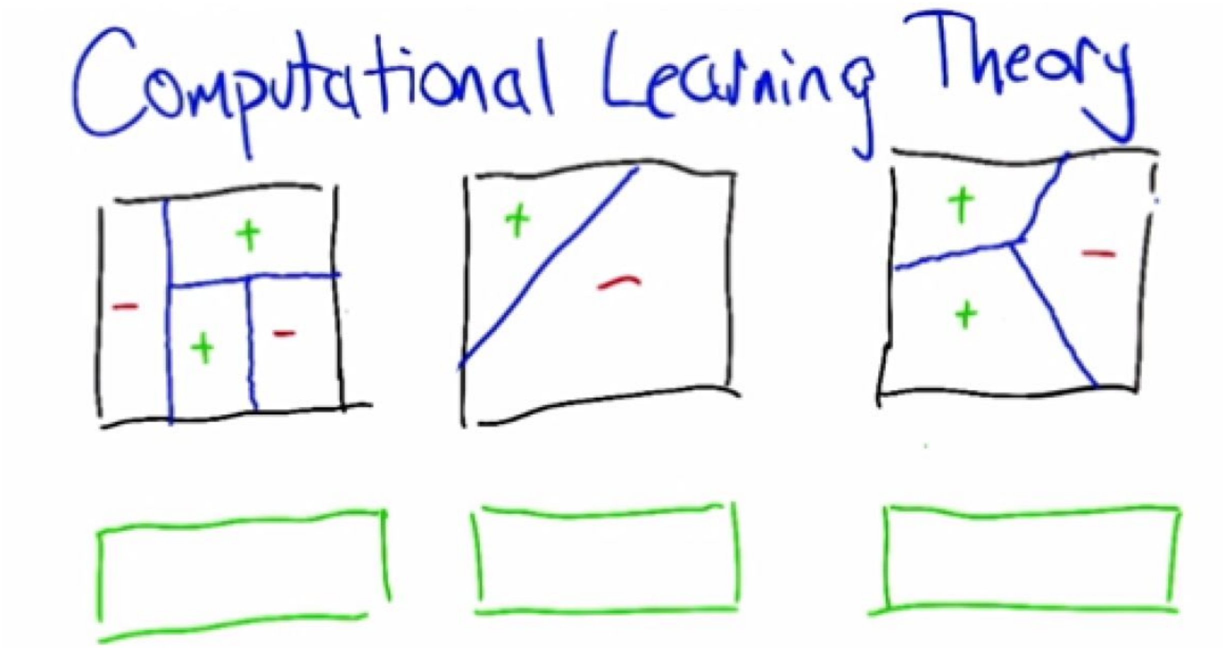
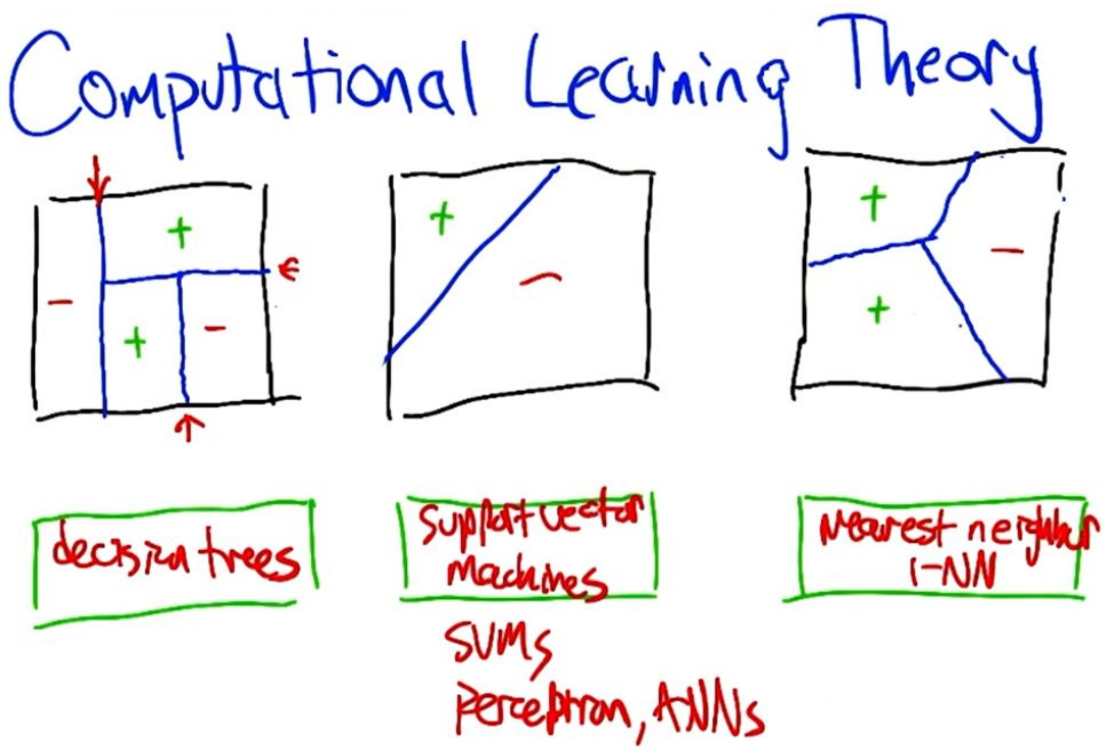
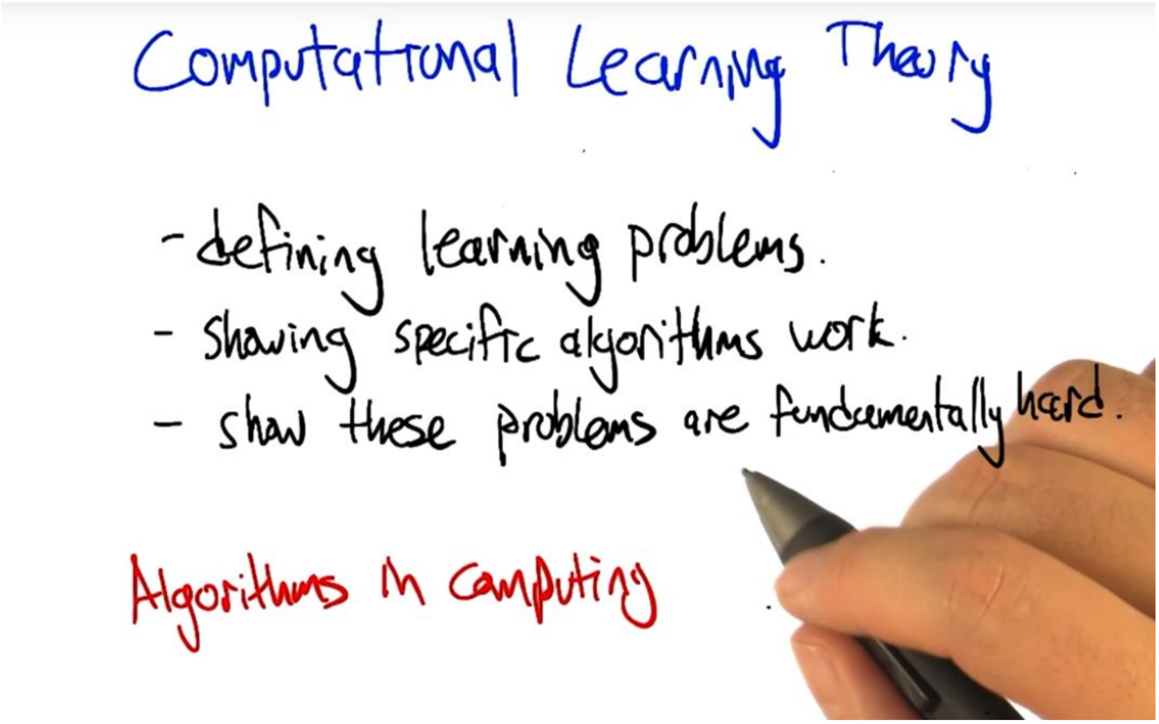
1. Quiz Computational Learning Theory



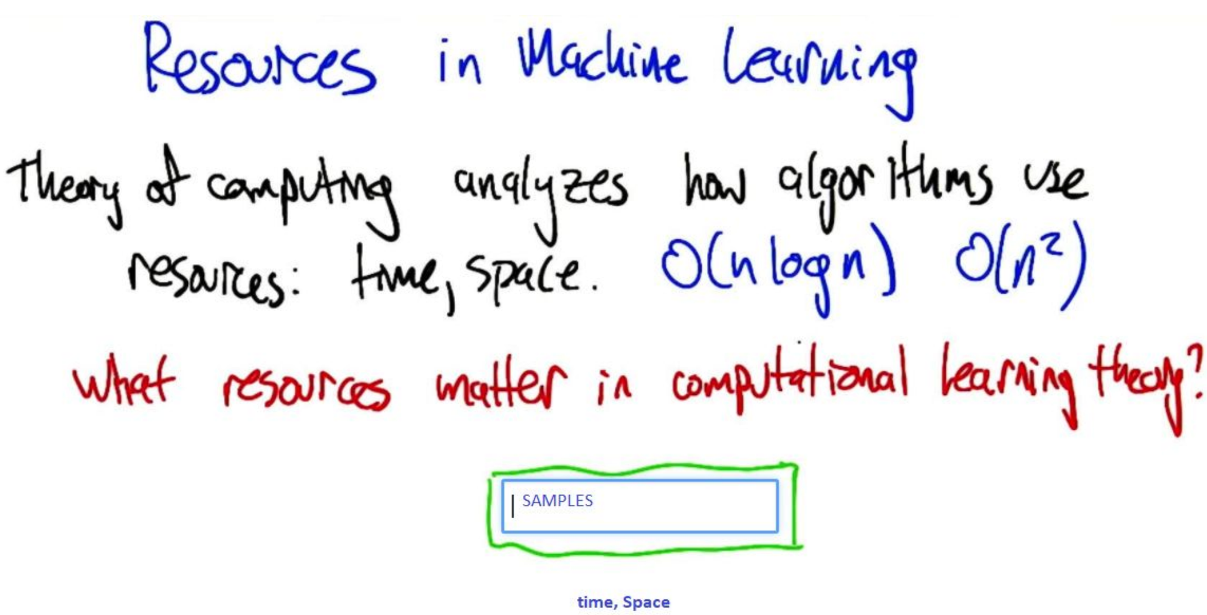
Answer:



2. Learning Theory



3. Quiz: Resources in Machine Learning



Answer: Samples (Time, Space)

4. Defining Inductive Learning

Defining Inductive Learning

Learning from examples

1. Probability of successful training $1-\delta$
2. Number of examples to train on m
3. Complexity of hypothesis class complexity of H
4. Accuracy to which target concept is approximated. ε
5. Manner in which training examples presented. batch/online
6. Manner in which training examples selected.

5. Selecting Training Examples

Selecting Training Examples

Learner / Teacher

1. Learner asks questions of teacher.

   $c(x)$? Learner

2. Teacher gives examples to help learner.

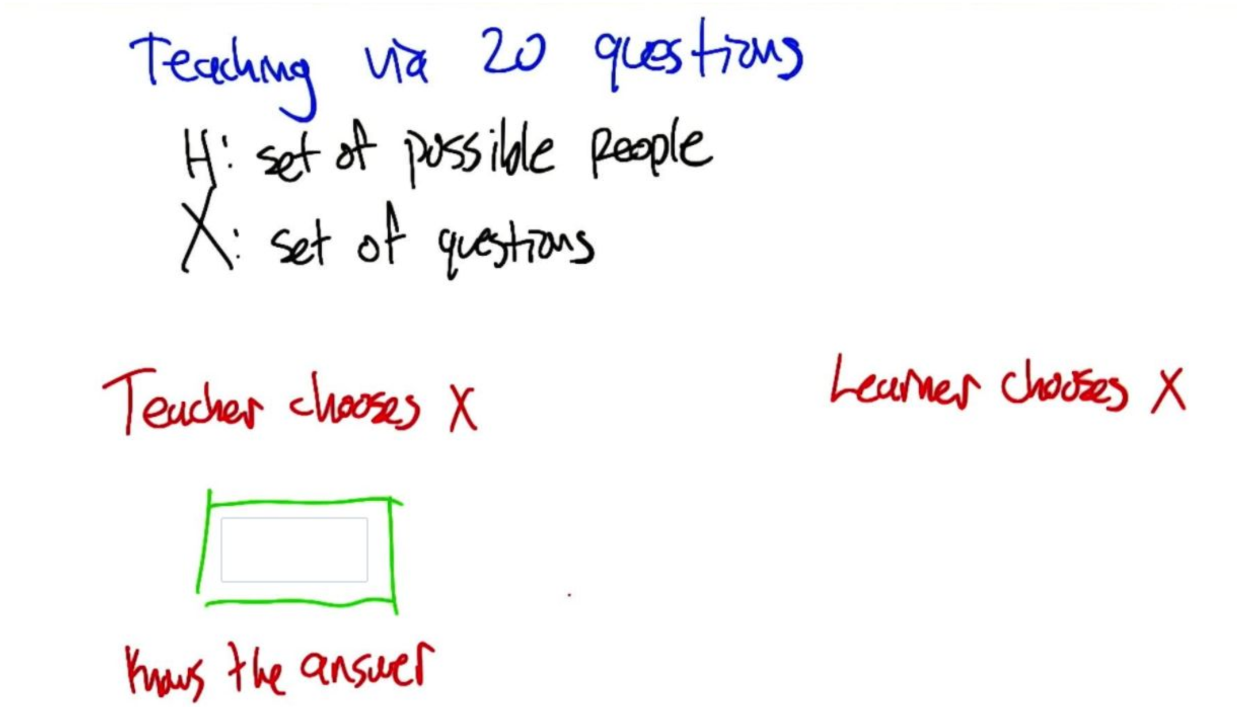
   Teacher chooses x, tells $c(x)$.

3. Fixed distribution

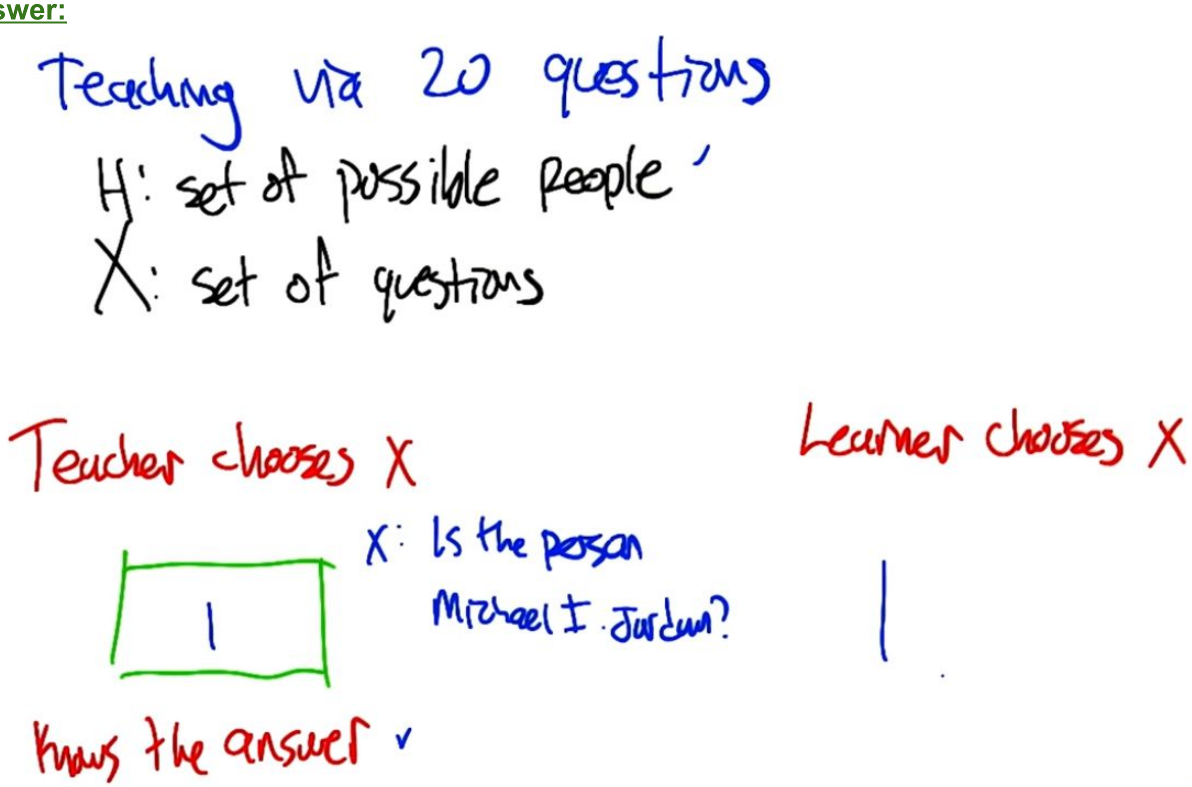
   x chosen from D by nature.

4. Evil- worst distribution

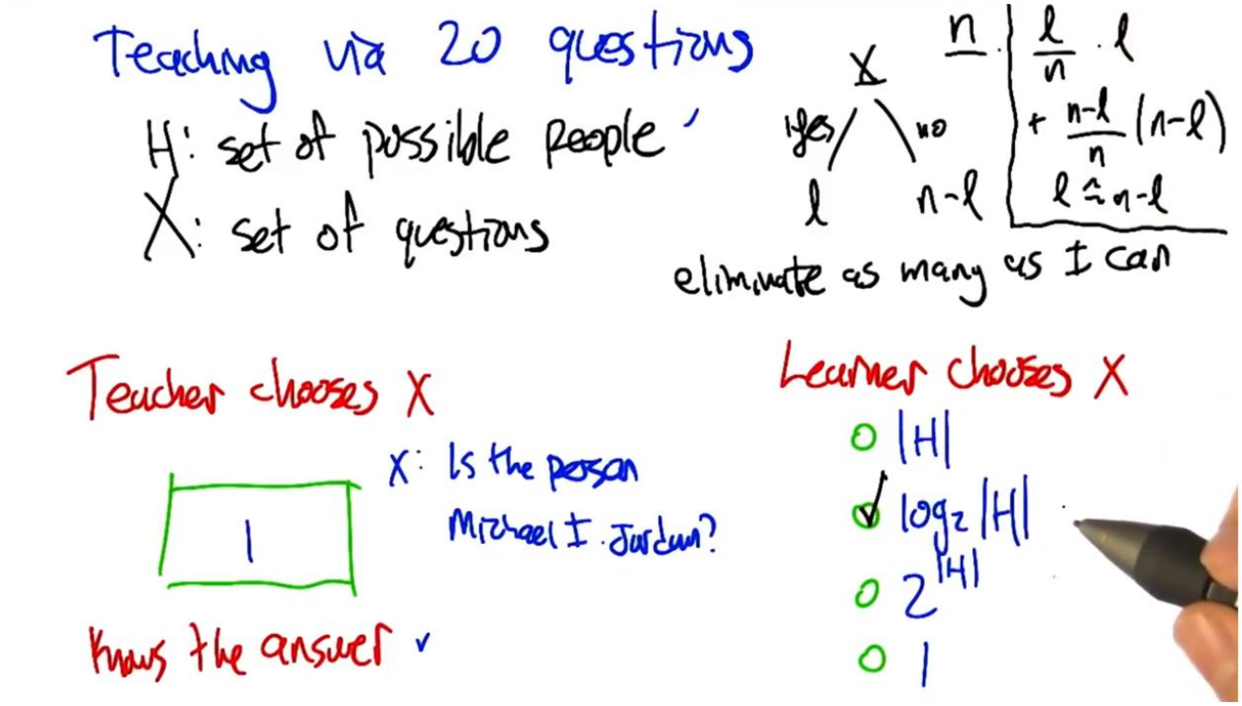
6. Quiz: Teaching Via 20 Questions



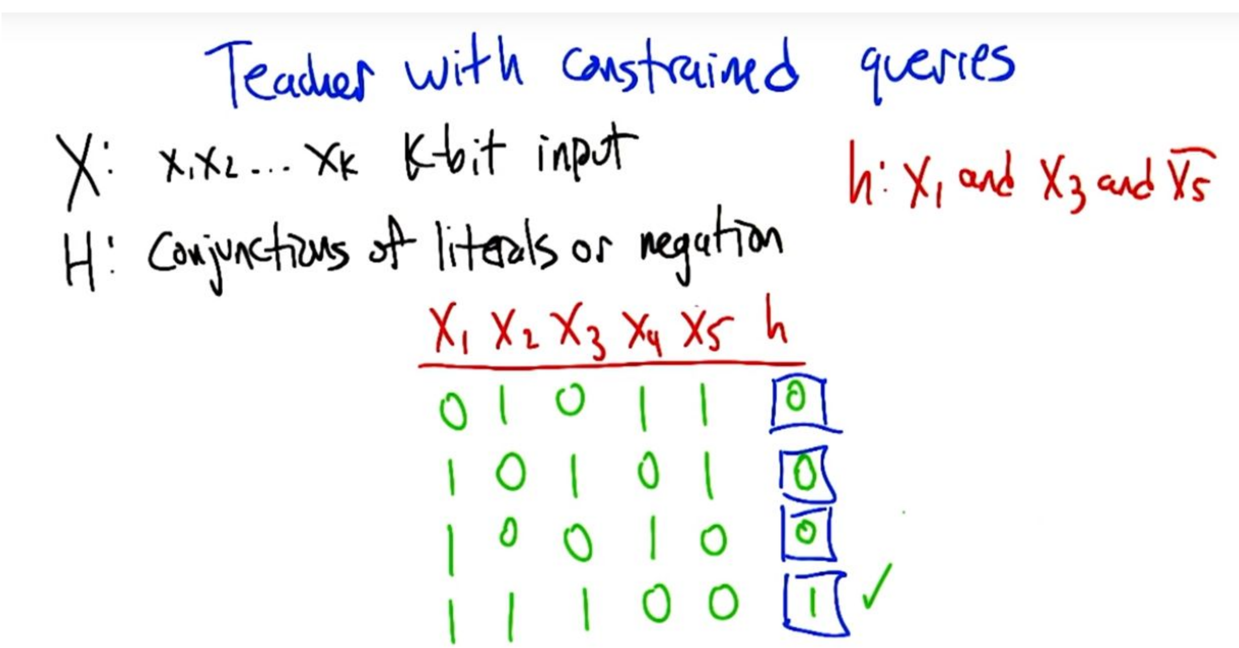
Answer:



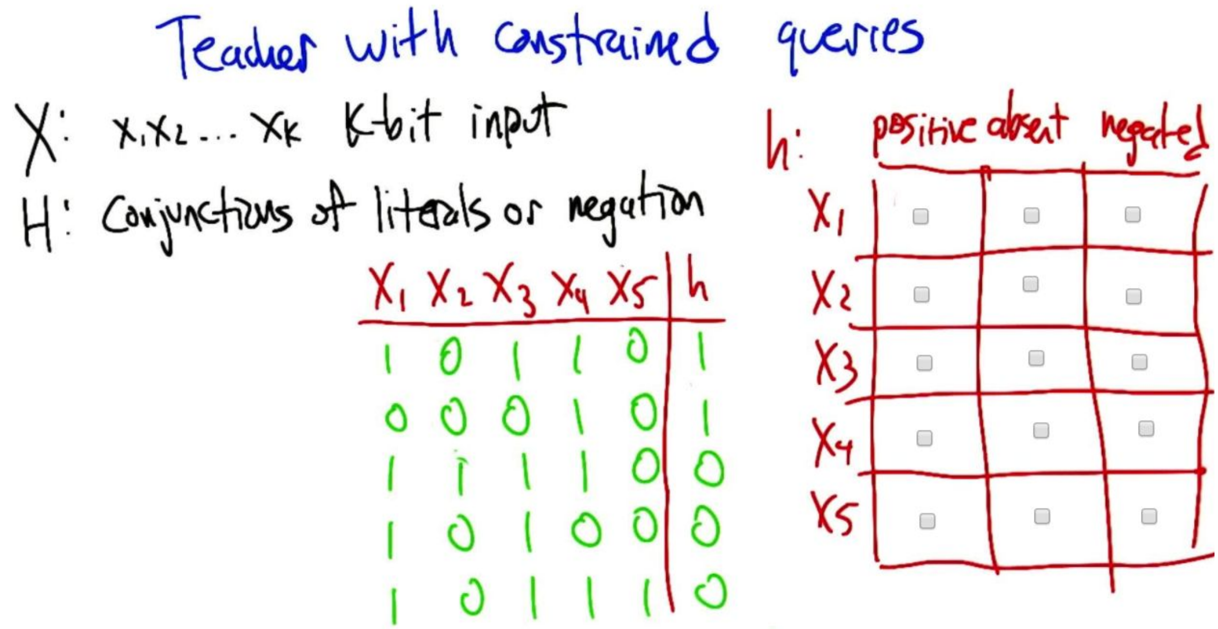
7. Quiz: The Learner



8. Teaching With Constrained Queries



9. Quiz: Reconstructing Hypothesis



M: All right, so here, here is, here's a table of examples with the input pattern and the actual output pattern of the hypothesis we're looking for. But I'm not going to tell you the hypothesis, you have to figure it out. So the way you're going to do that is, by figuring out for each variable, does it appear in the conjunction in its positive form, in its negative form or it's not there at all? For example if you want to say the hypothesis is X1 and not X5, you would write something like X1 and not X5, and the other ones are all absent. And just to be clear Charles, I, I've picked this particular set of examples so that, you particularly, Charles Isbell PhD, would have the best chance of figuring out the hypothesis with the smallest number of examples.

C: I appreciate that.

M: Go.

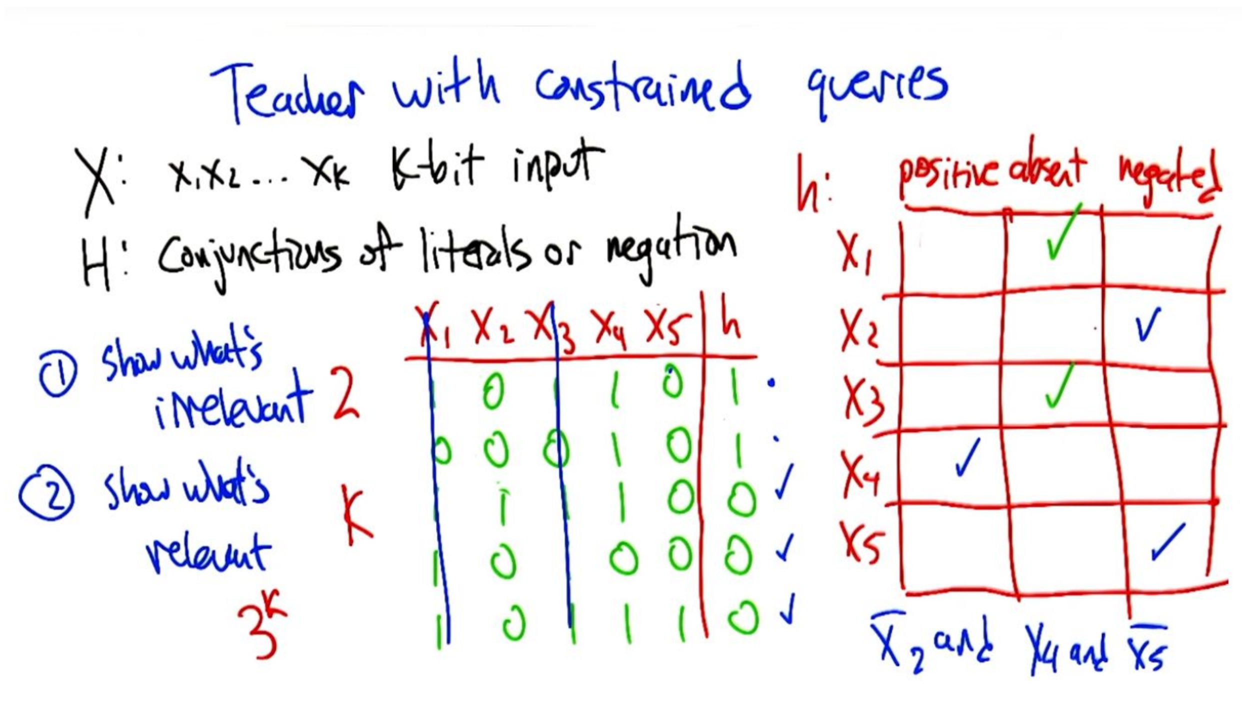
ANSWER:

M: So, what, how do you start with this now, Charles?

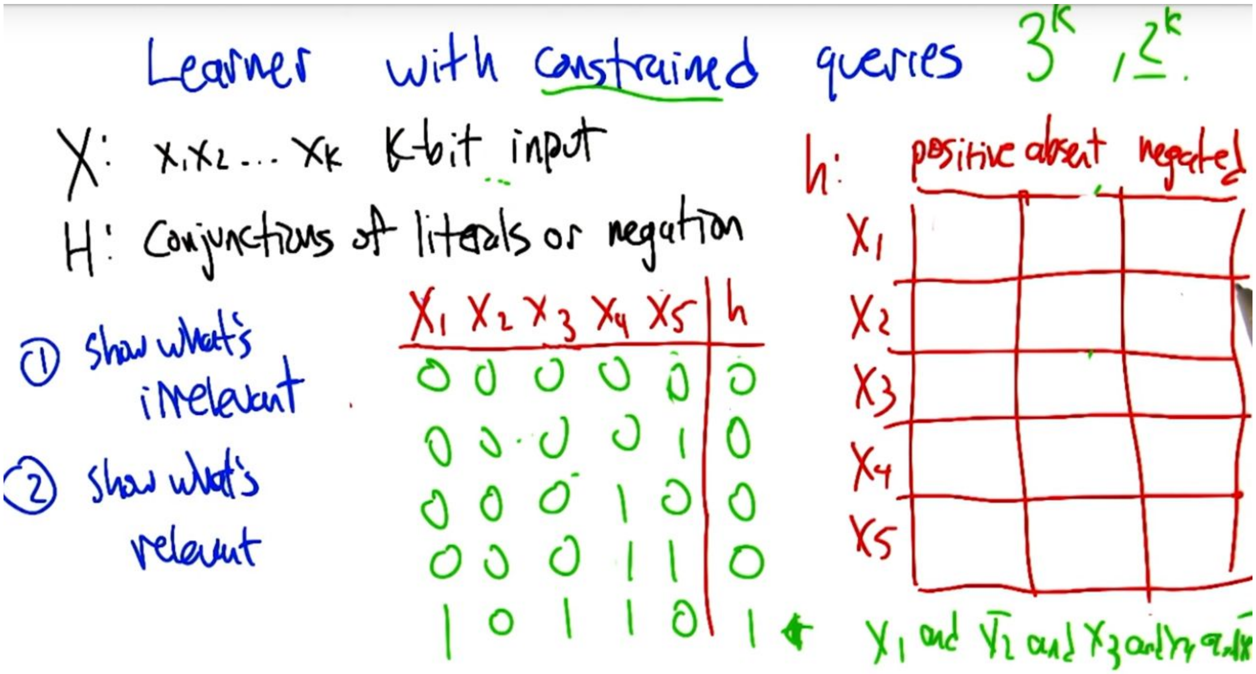
C: Okay, so the first thing I'm going to do is, I'm going to look at the first two examples. And the reason I'm doing that is because I know they both generate true. And so I'm going to look for variables that are inconsistent. So if I look at X1, for example, it's a one in the first case, and a zero in the second case. So it can't be the case that it's required in order for this to be true. And the same would be true for x3. So I'm going to say that neither x1 nor x3 matter. By contrast, x2, x4, and x5 all have the same values in both of those cases.

M: So we don't know much about them quite yet.
C: No
M: But let's so let's, that seems very well reasoned. So we know that x1 can't be part of this formula and x3 can't be a part of this formula. So, let's just sort of imagine
that they're not there cause they don't really give us any information any more.
C: Beautiful.
M: Alright, So what's left?
C: So what's left is now to make certain that x2, x4 and x5 are necessary, and particularly necessary with the, the values that they have. So I guess all I can really do is see if there's anything else to eliminate. If I were just looking at the first two, I would think that the answer was not X2, X4, not X5.
M: Alright. so, hang on, not X2, X4, not X5.
C: Right. So, that's what I currently think it is based upon what I just saw.
M: And that would be, that's consistent with the first two examples.
C: Right. And so, now I want to make certain is consistent with the next three examples. This is the easiest way for me to think about this, anyway. So, let's see. Not X2, X4, so that should be false, which it is. They're all false, so let's see not X2. But x4, up, that should be false, which it is. And then I do that same thing. But wait, why isn't that the answer? That can't be the answer.
M: It is the answer, you got it.
C: Huh I got it right.
M: So the thing to notice is that in, in these first two examples we have X2 is false X4 is true and X5 is false and that's enough to make the conjunction true but making- flipping any one of those bits is enough to make it false so what I showed in the in the remaining examples is that just by turning this X2 into an X1 leaving everything else the same we lose it. Similarly if we flip the X4 to zero and leave everything else the same, we lose it. Similarly if we flip x5 to one, and leave everything else the same, we lose it. So that means that each of these is necessary to make the conjunction. They're all actually in there.
C: That's just what I was thinking. So, in other words, you gave me some positive examples to eliminate things that were necessary, and then you gave me negative examples to validate that each of the variables that I saw so far were necessary because getting rid of any one of them, gave me the wrong answer.
M: Exactly. Let's, let's even write down those two steps. So the first thing was show what's irrelevant. And how many questions How many queries might we have needed to show that?
C: Well, one per variable.
M: Well actually we only need two because what I did is I, I used, all the relevant ones I kept the same and all the irrelevant ones I flipped from one to the other. I just have to show you that it's still, the output is still one even though they have two different values.
C: Oh, no, no. When I said all of them, you know, k of them was because I didn't know that, what if all of them were irrelevant
M: Then it would still be two. because then I could just show you the all zeroes, and the all ones
C: You're right. You're right. You just need, oh that's right. That's exactly right.
M: Alright. And then I have to show you that the, that each, the remaining variables is relevant by flipping it and showing you that the answer is zero. And how many questions did I need to do

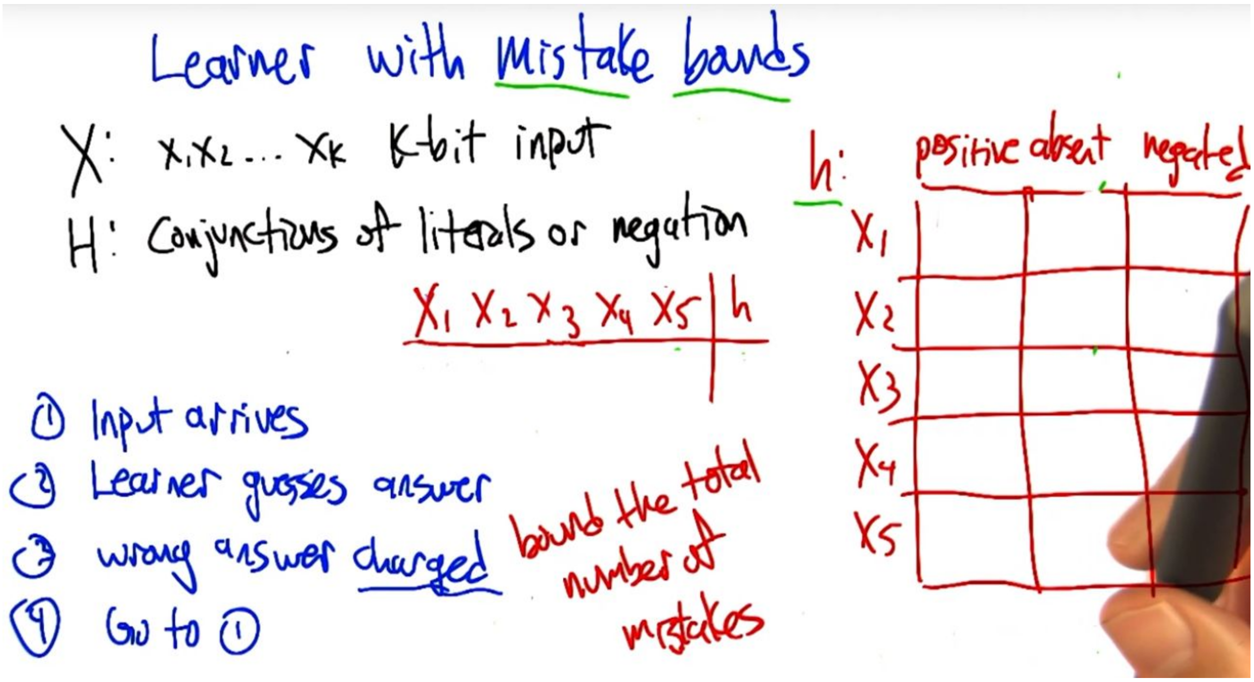
for that?
C: Three.
M: Yeah, three in this case cause there were three variables that were used in the formula. What's the most it could be?
C: Well k, cause all of them could be relevant.
M: Yeah, so it's you know, it's kind of interesting that, that in fact the total number of hypothesis here is three to the K. Because you know, you can see it right off this table that for each of the variables, it's either positive, absent or negated. But the number of questions that a smart teacher had to ask was more like, K plus two.
C: Huh.
M: Which is pretty powerful.
C: Right, so. The smart teacher can help me do this in linear term. So what if I were, I, I didn't have the teacher who could give me these examples and I always had to ask?
M: That's a good question, let's let's do that.



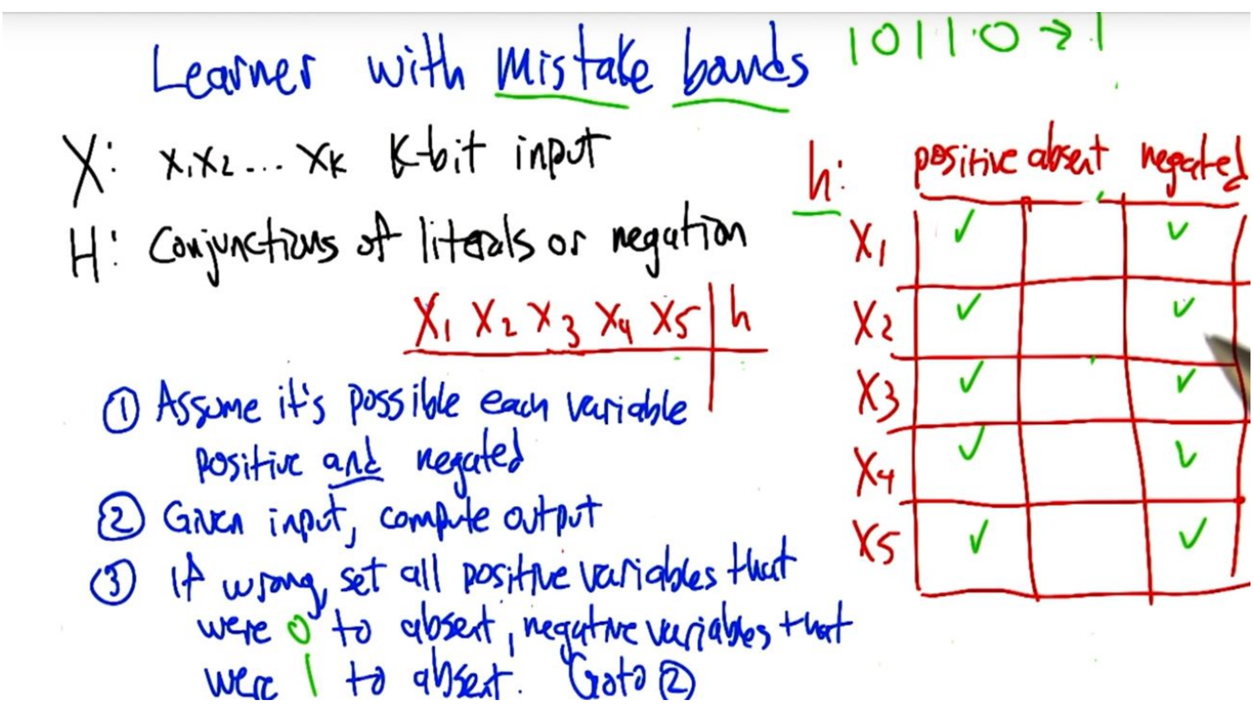
10. Learner With Constrained Queries

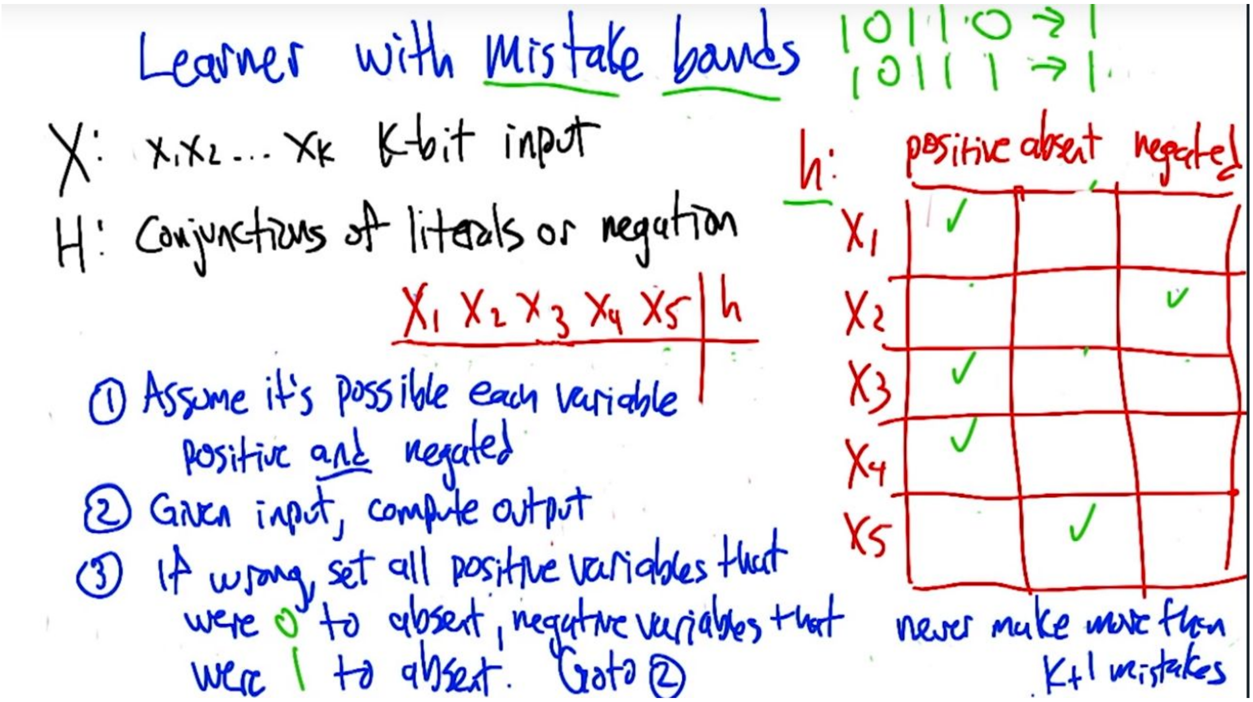


11. Learner With Mistake Bounds



Let’s turn this to algorithm:





12. Definitions

Definitions

learner chooses ✓
teacher chooses ✓
nature chooses
(mean teacher) ✓

Computational complexity
How much computational effort is needed for a learner to converge?

Sample complexity - batch
How many training examples are needed for a learner to create a successful hypothesis?

Mistake bounds - online
How many misclassifications can a learner make over an infinite run?

13. Version Spaces

Version Spaces

True hypothesis: $c \in H$

Training set: $S \subseteq X$

$c(x)\ \forall x \in S$

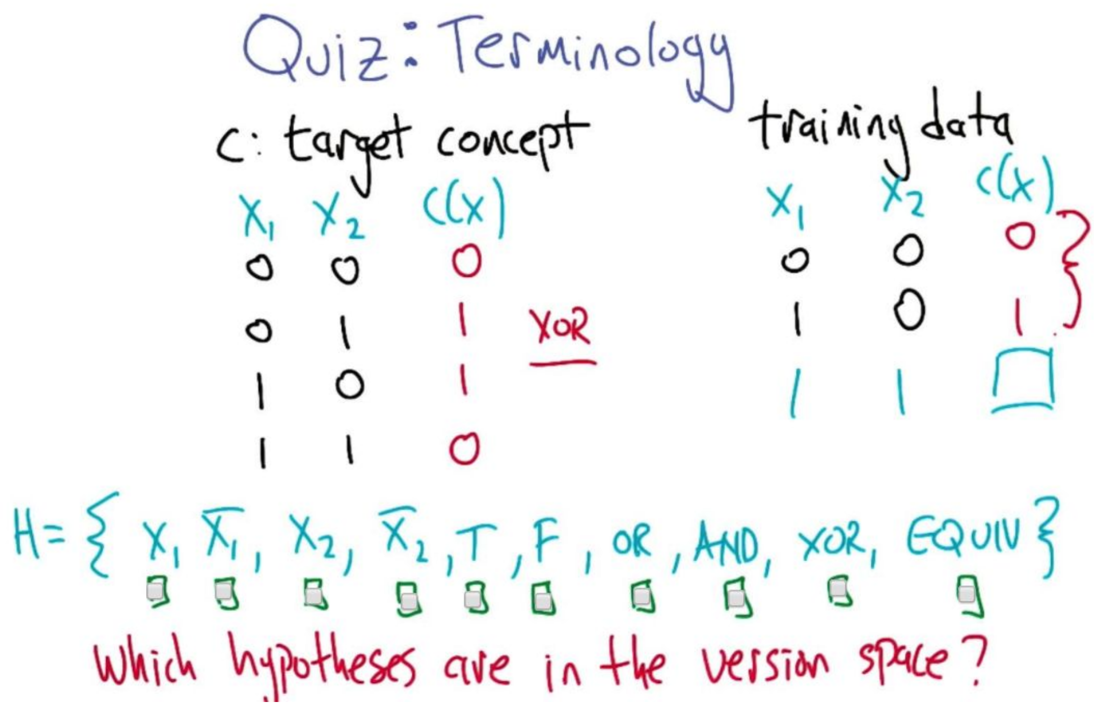
candidate hypothesis: $h \in H$

consistent learner: produces $c(x) = h(x)$ for $x \in S$

version space: $VS(S) = \{h \text{ s.t. } h \in H \text{ consistent wrt } S\}$

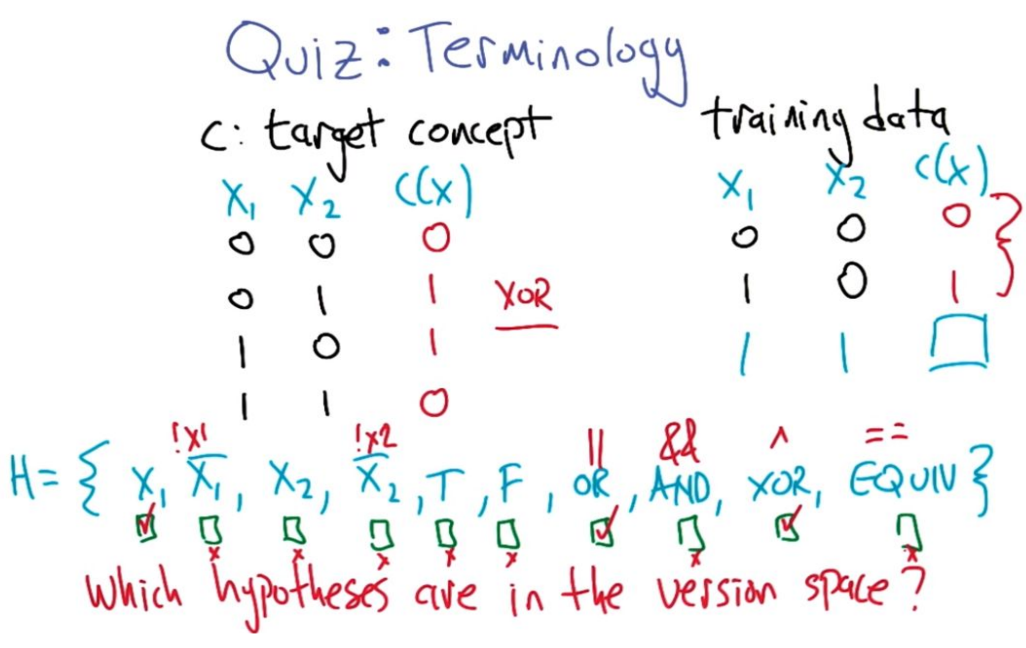
Hypotheses consistent with examples.

14. Quiz: Terminology



M: All right Charles, what do you think?

C: Okay, so being in the version space just means that you're consistent with the, data that you see. Right?



15. Error of h

PAC Learning - Error of h

Training error: fraction of training examples misclassified by h.

True error: fraction of examples that would be misclassified on sample drawn from D.

$$\text{error}_D(h) = \Pr_{x \sim D}[c(x) \neq h(x)]$$

16. PAC Learning

PAC Learning

C: Concept class

L: Learner

H: Hypothesis space

n: $|H|$, size of hypothesis space

D: distribution over inputs

$0 \leq \varepsilon \leq 1/2$ error goal

$0 \leq \delta \leq 1/2$ certainty goal $(1-\delta)$

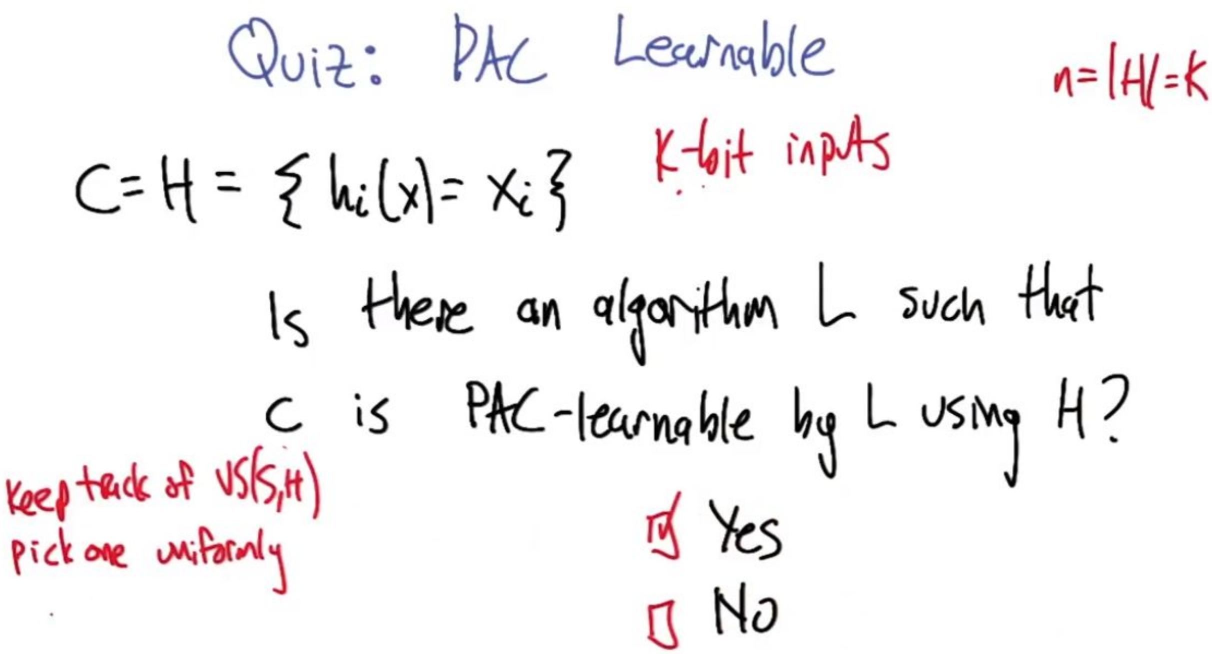
Probably ($1-\delta$) approximately (ε) correct! ($\text{error}_D(h) = 0$)

C is PAC-learnable by L using H iff learner L will, with probability $1-\delta$, output a hypothesis $h \in H$ such that $\text{error}_D(h) \leq \varepsilon$ in time and samples polynomial in $1/\varepsilon$, $1/\delta$, & n.

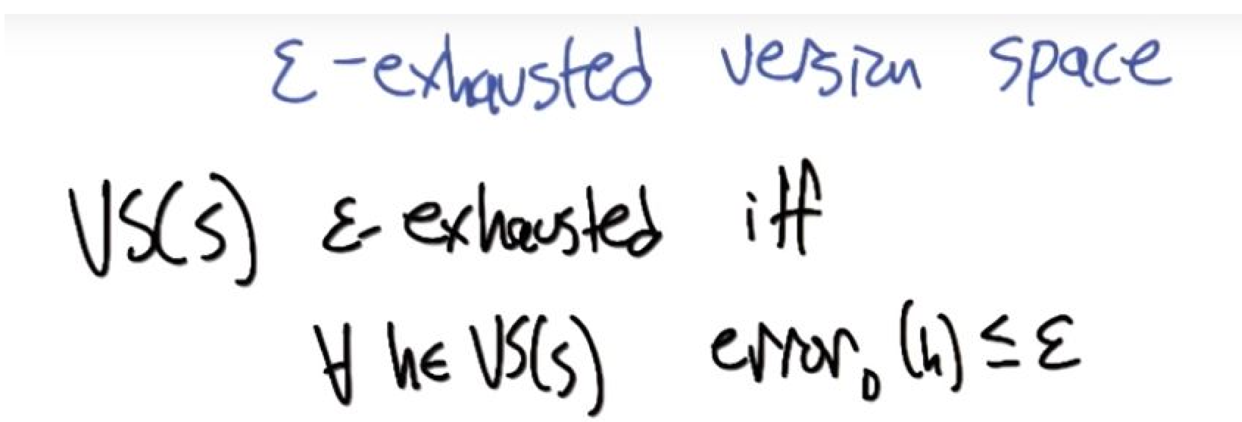
17. PAC Learning Two

M: Alright, now we can actually dive in and give a definition for PAC-learnable. So, a concept class, C, is PAC-learnable by some learning algorithm, L, using its own representation of hypothesis, H, if and only if that learner will, with high probability, at least from its set of hypothesis. That has error less than or equal to epsilon, so it's very accurate. And it needs to be the case that the time that it takes to do this. And the number of samples that it needs draws
from this distribution D is relatively small. In fact, bounded by a polynomial in one over epsilon, one over delta and the size of the hypothesis space n.
C: Okay, so in other words you're saying something is PAC-learnable if you can learn to get low error, at least with some high confidence you can be fairly confident that you will have a low
error in time that's sort of polynomial in all the parameters.
M: Exactly, and you can see here that this, this epsilon and delta actually giving us a lot of wiggle room, if you really want to have perfect error. Or perfect certainty. Then these things go to infinity. So, you just, you need, you need to look at all the possible data.
C: Mm.
M: So, yeah this is really going to only give us partial guarantees.
C: Okay. Sure. Okay, I think I understand that. .

18. Quiz: PAC Learnable



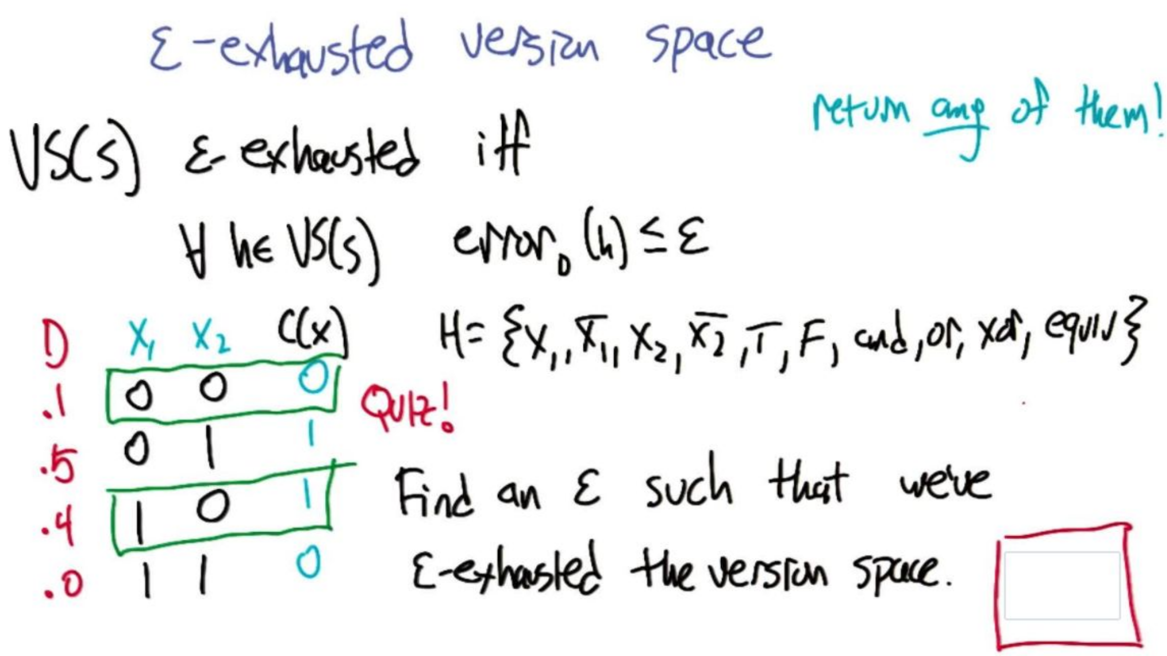
19. Epsilon Exhausted



C:what you're saying is, something is epsilon exhausted, a version space is epsilon exhausted exactly in the case when everything that you might possibly choose has an error less than epsilon.
M: Sure.
C: Period. And so if there's anything in there that has error greater than epsilon, then it's not

epsilon exhausted.
M: Right. It's still epsilon energized.
C: Right. That makes a lot of sense, epsilon energized. That's a pretty good name for a band. OK.

20. Quiz: Epsilon Exhausted



Here's our example from before. We've got our target concept, which is X or, we've got our training data which is these, with the things that are in the green boxes here, 0 0 output 0 1 0 output 1. And this time, I'll actually write down a distribution D that gives the. Probability of drawing each of these different input examples. All right? So now what we'd like to do is find an epsilon such that that this training set that we've gotten has epsilon exhausted the version space.

Answer:

M: Okay, so you saying the ones that are in green are the training examples that we see, right?
C: Right.
M: So we should be able to use that to figure out what the version space actually is.
C: Yeah, I think that's right. Okay, so, then what we can do is given that those are the three things that we've done. We could actually compute, what the error is according to this distribution for each of those three.
M: Yes, exactly so.
C: So let's, let's start with X1. So which one, x1, so all three of those are going to get the first one and the third one correct, right?

M: All of them are going to get the first one and the third one correct. Yes, by design.
C: By design.
M: Right, to be in the version space.
C: So now we can ask which ones will get the second one wrong? The fourth one doesn't matter because it has zero probability of showing up.
M: That's right. So, it doesn't matter if you get this one right or wrong, it's not going to contribute to this true error measure.
C: Okay. So let's look at x one. So x one will in fact get the second one wrong, because the output Is not the same as the value for x one.
M: Good. And so what;s the probability that x one, this hypothesis x one, is going to give a wrong answer on a randomly drawn input?
C: Well, half the time it will get the second answer, and so the error is, in fact, one half.
M: Yes. Exactly. Good. All right. Let's move on to the or.
C: Okay. So, we can do an easy one actually. We can do xor. Since we know xor is the right answer, we know It will has a probability of being wrong of zero.
M: Oh, good point.
C: Okay. And so for or we can do the same thing. So is, we know it's going to get the first and the third ones right. So now we can ask whether it's going to get the second one, right. And zero or one is in fact true. Or one. So in fact it also has an error of zero.
M: Okay.
C: Which is kind of interesting. So and so, even though the function is xor, if we can get to the point where we have or or xor left, we actually will get zero true error.
M: That's right.
C: But in the meantime, because x1 has still survived the two examples that we have. Epsilon is therefore 0.5.
M: Right, in particular, we're saying that, this is, if, if epsilon were smaller than 0.5, then it wouldn't be epsilon exhausted because you'd have a hypothesis that has error that's too high.
C: Right.
M: So this is the smallest epsilon that we can use. And in fact, we let you through if if it was anything value that I was really hoping you'd be able to reason out.
C: Okay, well that all made sense.
M: Good, nice work.

ε-exhausted version space

return any of them!

VS(S) ε-exhausted iff

$\forall h \in VS(S) \quad error_D(h) \leq \varepsilon$

$error_D(h)$ .5

$H = \{x_1, \bar{x}_1, x_2, \bar{x}_2, T, F, \text{and}, \text{or}, \text{xor}, \text{equiv}\}$

D	x_1	x_2	$c(x)$
.1	0	0	0
.5	0	1	1
.4	1	0	1
.0	1	1	0

Quiz!

Find ~~an~~ the smallest ε such that we've ε-exhausted the version space.

$\boxed{.5}$

21. Haussler Theorem

Haussler Theorem - Bound True Error

Let $error_D(h_1, \ldots, h_k \in H) > \varepsilon$ High true error.

How much data do we need to "knock out" these hypotheses?

"low" probability of match

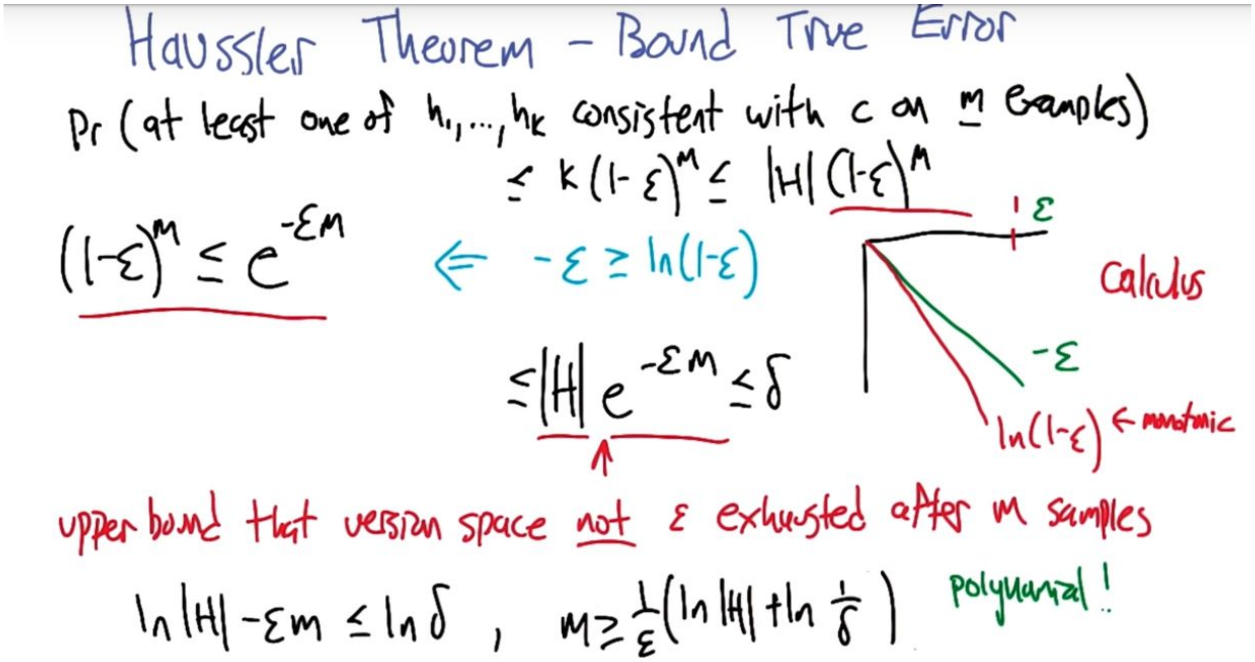
$\Pr_{x \sim D}(h_i(x) = c(x)) \leq 1 - \varepsilon$

$\Pr(h_i \text{ consistent with } c \text{ on } \underline{m} \text{ examples}) \leq (1-\varepsilon)^m$ independent and rand+...

$\Pr(\text{at least one of } h_1, \ldots, h_k \text{ consistent with } c \text{ on } \underline{m} \text{ examples})$

$\leq k \cdot (1-\varepsilon)^m \leq |H|(1-\varepsilon)^m$

22. Haussler Theorem Two



23. Quiz: PAC Learnable Example

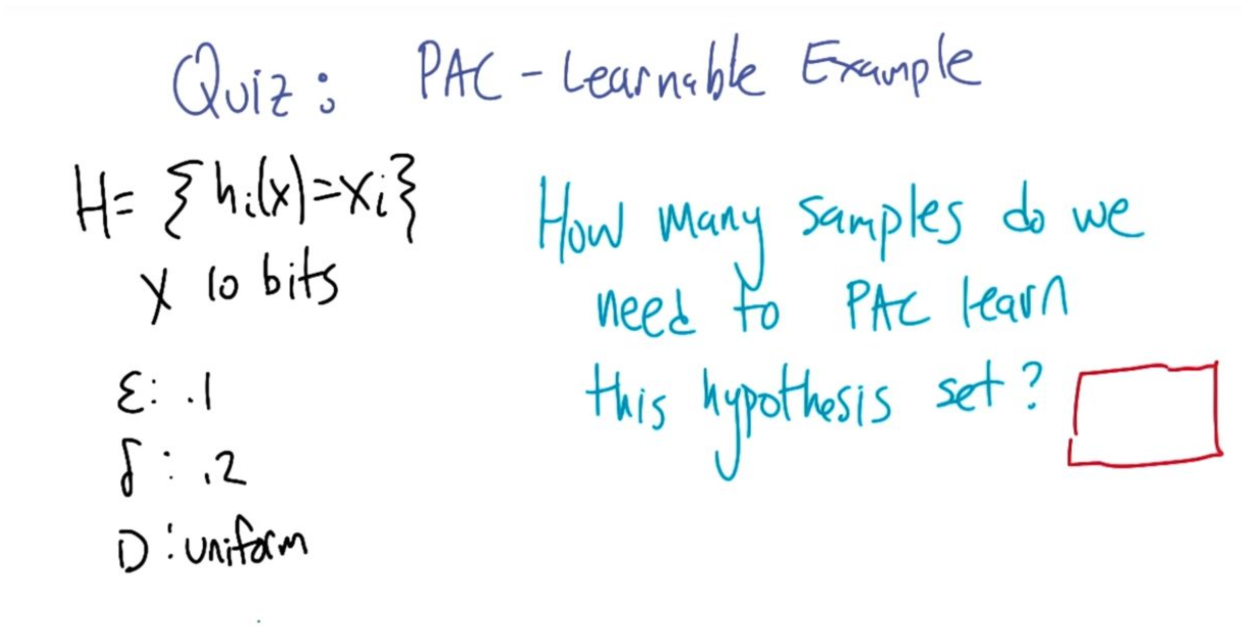
M: This puts us in a good position to be able to go back to that question we looked at before, so let's look at it a little bit differently this time.

If our hypothesis space is the set of functions that take 10 bit inputs and there is a hypothesis corresponding to returning each of those 10 bits, separate hypothesis, one returns the first bit, one returns the second bit. And so on, and now we have a target of Epsilon equals point 1, so we would like to, return a hypothesis whose error is less than or equal to point 1, and we want to be pretty sure of it, the error, or failure probability needs to be less than or equal to point that our distribution of our inputs is uniform

M: So given this setup, how many samples do we need to pack learn this hypothesis set.

M: And remember the algorithm that we're going to use is we're going to draw sample of the size that we want. Then we are going to be confident that we've epsilon exhausted that, that, version space. And so anything they left in the version space should have low error. And that procedure should fail with probability, no more than .2.

C: Right. So it's just exceeds probability .8.



ANSWER:

C: Alright, so M is greater than or equal to, 1 over epsilon times the natural log of the size of the hypothesis space.

M: Mm, which is what, that is not one of our variables here.

C: ten.

M: Yeah. Right, so it's not 2 to the 10. Even though the input space is 2 to the 10. The number of hypotheses. There's one hypothesis corresponding to each of the bit positions. So, good?

C: Right. Plus, the natural log of 1 over delta. So that would be greater than or equal to ten times the natural log of ten.

C: Plus the natural log of, five. So, let's see. The natural log of ten is something like three point something, the natural log of five is something like, two point something. We add those up, multiply by ten you're going to end up with 39.12.

M: Good, so, we need, you know, 40 samples?

C: Yeah. That sounds about right.

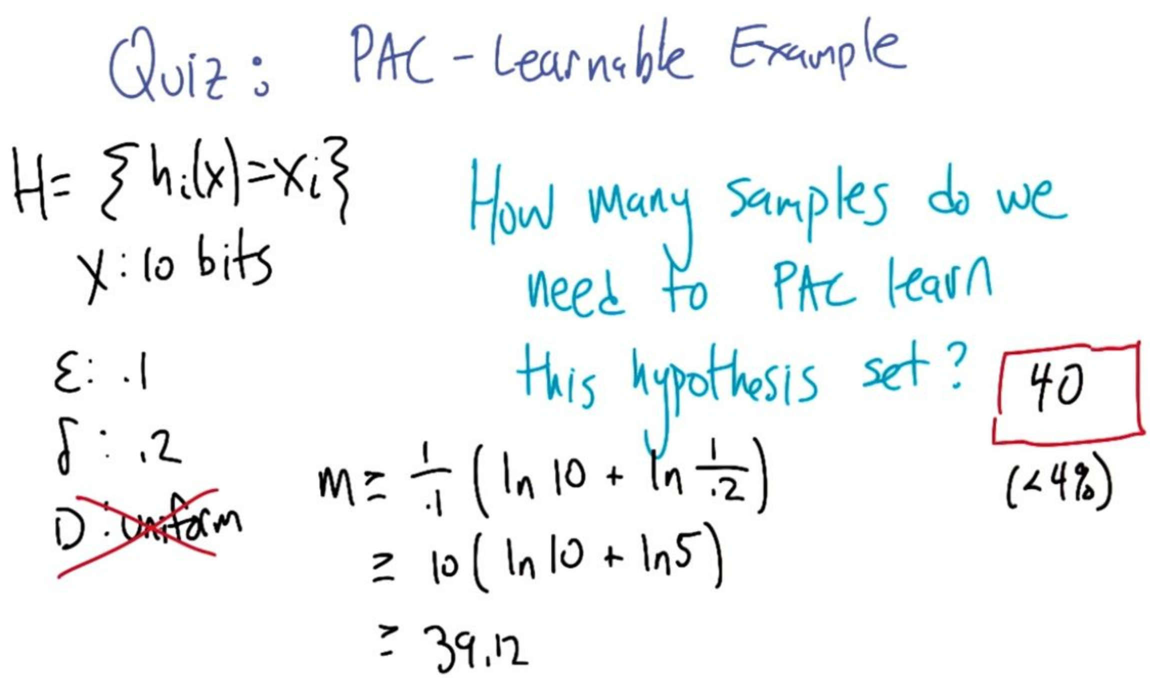
M: That actually doesn't sound too bad. Well, you know, it's not learning a very hard problem, but it's, you know, a pretty big input space. So let's see. What, how big is the input space? It's like two to the ten, which is.

C: 1,024.

M: 1,024. So how much of 1024 is 40? It's, it's, you know, less than 4%. Hm, that's not bad.

C: Before we leave this quiz, let me point out one more thing: that this bound is actually agnostic to the distribution from which samples came, so this idea that it's from a uniform distribution is actually not being directly used here. So so this is pretty cool. It actually doesn't matter, we only

need 40 samples no matter what the distribution is. It's not like some distributions are harder or easier ,because we are measuring the true error on the same distribution that we used to, to create the training set. So if it's a really hard distribution and some tough examples never appear, then we're unlikely to see them in the training set, but they're not going to contribute very much to the true error.
M: Well that makes sense. So the distri, oh right. So in some sense, I
mean, I guess the equation doesn't show this, but in some sense, the distribution is, cancels out between the training and the true error
C: Yeah, that's one way to think about it.
M: Well, I like that. So 40 is pretty good to get 10% error. If we wanted to get say, only 1% error,
Then we would go from 40 to 400.
C: That's a good point, yeah.
M: And it's, it's, it's one decimal point even. And so, that would be about 40% of the data.
C: Yeah, that's true. Yeah, if we want to go a little bit beyond that we may need all the data multiple times.
M: Mm-hm.
C: Yeah, but this example doesn't look so bad. So let's just move on before we think about it too hard.
M: Okay. That seems fair, I like that.



24. What Have We Learned