Introduction to Machine Learning

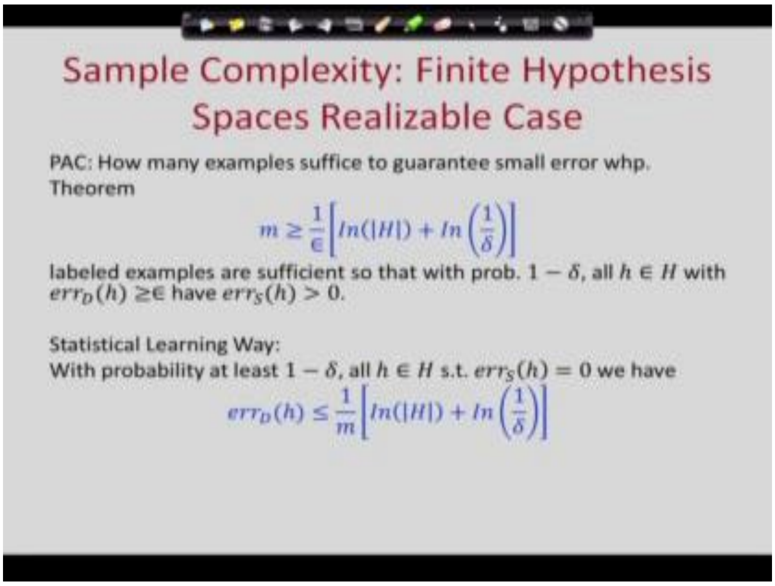
Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module - 7
Lecture - 33
Sample Complexity: Finite Hypothesis Space

Good morning, in the last class we looked at computational learning theory and we looked at situations where a learner sees a certain number of examples, outputs the consist hypothesis and we derived bounds on how many examples the learner has to see, so that a consistent hypothesis is a PAC hypothesis.

So, today we will continue our discussion and look at the situation where the hypothesis output by the learner has non-zero error on the training set and then we will look at a few worked out examples.

(Refer Slide Time: 01:03)



Sample Complexity: Finite Hypothesis Spaces Realizable Case

PAC: How many examples suffice to guarantee small error whp.
Theorem

$$m \geq \frac{1}{\epsilon}\left[ln(|H|) + ln\left(\frac{1}{\delta}\right)\right]$$

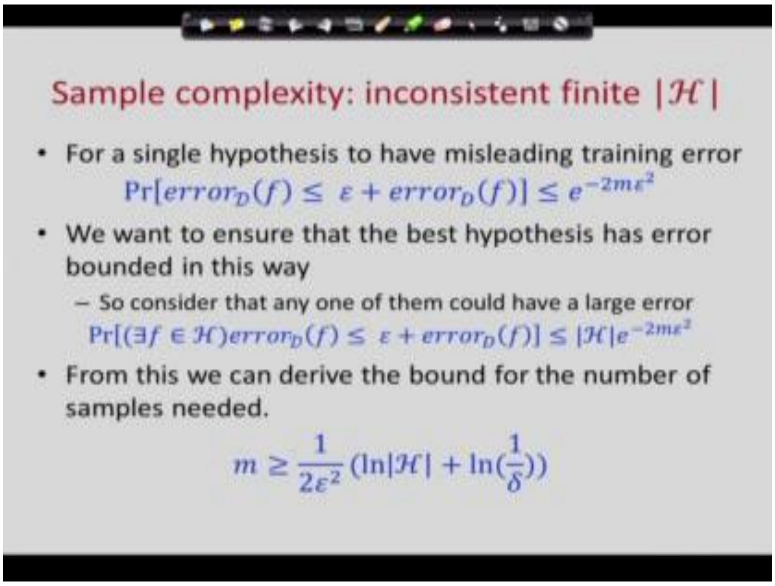
labeled examples are sufficient so that with prob. $1-\delta$, all $h \in H$ with $err_D(h) \geq\epsilon$ have $err_S(h) > 0$.

Statistical Learning Way:
With probability at least $1-\delta$, all $h \in H$ s.t. $err_S(h) = 0$ we have

$$err_D(h) \leq \frac{1}{m}\left[ln(|H|) + ln\left(\frac{1}{\delta}\right)\right]$$

So, in the last class we have seen that in the finite hypothesis case where the learner outputs a consistent hypothesis what value of m is required.

(Refer Slide Time: 01:18)



Sample complexity: inconsistent finite $|\mathcal{H}|$

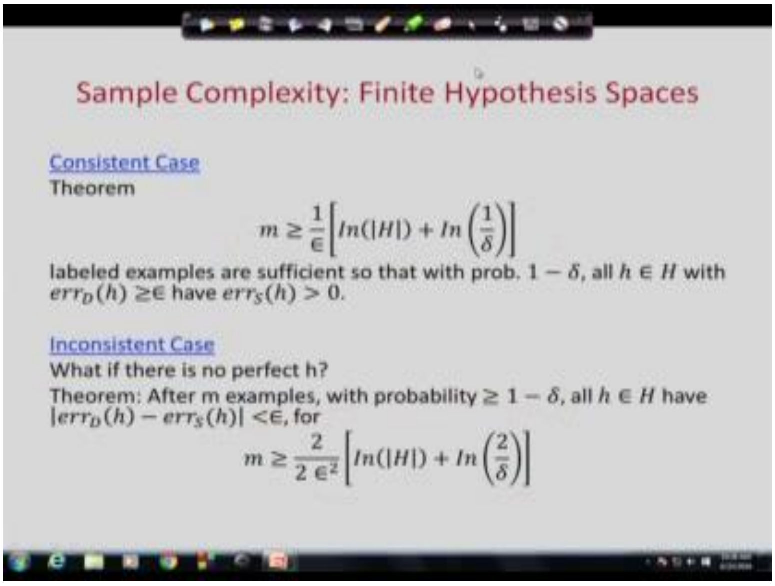
- For a single hypothesis to have misleading training error
  $$Pr[error_D(f) \le \varepsilon + error_D(f)] \le e^{-2m\varepsilon^2}$$
- We want to ensure that the best hypothesis has error bounded in this way
  - So consider that any one of them could have a large error
  $$Pr[(\exists f \in \mathcal{H})error_D(f) \le \varepsilon + error_D(f)] \le |\mathcal{H}|e^{-2m\varepsilon^2}$$
- From this we can derive the bound for the number of samples needed.
  $$m \ge \frac{1}{2\varepsilon^2}(\ln|\mathcal{H}| + \ln(\frac{1}{\delta}))$$

Now, today we will look at how many samples will be required to guarantee a PAC hypothesis and we will explain, what we mean by this? Where the learner outputs a hypothesis which has some error, there is a sample error which is output by of the output hypothesis. Now, what we want to do is that we want to find the guarantee that the actual error, the true error of the hypothesis f over the distribution D (f) is a hypothesis which is output by the learner, which may have an error given by error D (f), where D is the sample, this is the sample error and this is the true error. So, we want the probability that the true error is less than equal to the sample error plus epsilon.

We have earlier seen that normally the true error will be greater than equal to the sample error, and we want that the true error is not so much different from the sample error, that is true error is less than equal to epsilon plus sample error for a given epsilon and we will find out that. If m is the number of training examples with respect to which you have tested, you have come up with the hypothesis then this probability is less than equal to e to the power minus 2 m epsilon square, and we want to ensure that the best hypothesis that our algorithm outputs has error bounded in this way. So, we consider that any one, suppose the hypothesis that the learner comes up with has an error, error D (f), this is the sample error

And we want to find out the probability that there exists any hypothesis whose true error is less than equal to this is less than equal to h times e to the power minus 2 m epsilon square. This is for a single hypothesis, this is where they exists any hypothesis at all and from this we can derive the bound by algebraic manipulation that in order to ensure that the number of examples required must be greater than equal to 1 by 2 epsilon square times logarithm h plus logarithm 1 by delta. So, if you have this many numbers of examples, then the hypothesis which comes up with the training error of x will correspond to a hypothesis whose true error is less than equal to epsilon plus x.

(Refer Slide Time: 04:23)



Sample Complexity: Finite Hypothesis Spaces

Consistent Case

Theorem

$$m \geq \frac{1}{\epsilon}\left[ln(|H|) + ln\left(\frac{1}{\delta}\right)\right]$$

labeled examples are sufficient so that with prob. $1 - \delta$, all $h \in H$ with $err_D(h) \geq \epsilon$ have $err_S(h) > 0$.

Inconsistent Case

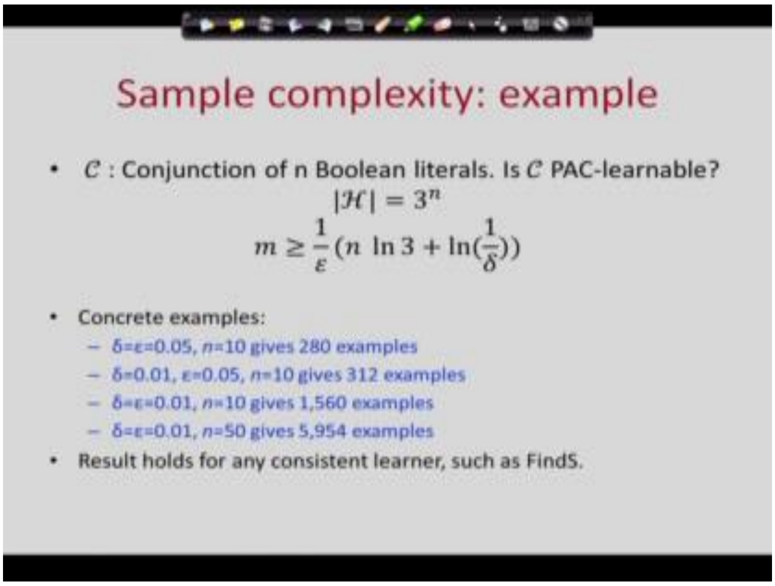
What if there is no perfect h?

Theorem: After m examples, with probability $\geq 1 - \delta$, all $h \in H$ have $|err_D(h) - err_S(h)| < \epsilon$, for

$$m \geq \frac{2}{2\,\epsilon^2}\left[ln(|H|) + ln\left(\frac{2}{\delta}\right)\right]$$

So, just to summarize, in consistent case we found this bound on m, this value of m in the inconsistent case we find this value of m and we can compare these two equations. So, here in the consistent case we came up with m greater than 1 by epsilon log h plus log 1 by delta, here we come up with this value of m. So, we see that there is a lot of similarity between these relations, but in this case, in the inconsistent case you require larger number of examples, larger by a factor of 2 by epsilon in order to give this guarantee, what is the guarantee? The guarantee is that the true error is no more than epsilon error epsilon plus sample error.

(Refer Slide Time: 05:32)



Sample complexity: example

- $\mathcal{C}$: Conjunction of n Boolean literals. Is $\mathcal{C}$ PAC-learnable?

$$|\mathcal{H}| = 3^n$$

$$m \geq \frac{1}{\varepsilon}(n \ \ln 3 + \ln(\frac{1}{\delta}))$$

- Concrete examples:
  - δ=ε=0.05, n=10 gives 280 examples
  - δ=0.01, ε=0.05, n=10 gives 312 examples
  - δ=ε=0.01, n=10 gives 1,560 examples
  - δ=ε=0.01, n=50 gives 5,954 examples
- Result holds for any consistent learner, such as FindS.

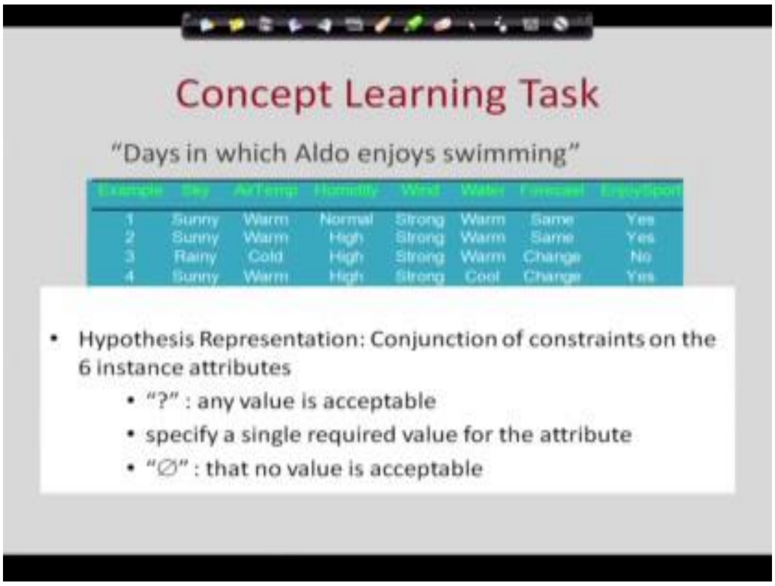
Now, let us look at an example of the first case. So, suppose we look at a particular hypothesis class. So, our hypothesis class is conjunction of Boolean literals, these are also called monomials. Suppose, x 1, small x 1, small x 2, small x n are your features these are your features. So, examples of Boolean literals are x 1 and x 2bar and x 4 x 2 and x 3 bar and x 4 bar x 2 bar and x 7, these are examples of conjunction of literals.

Now, this is our hypothesis space, what is the size of this hypothesis space? There are n literals in a particular hypothesis that literal may occur in the positive form, may occur in the negative from or may not occur at all. So, each for each literal there are three ways of it appearing in this formula and there are n literals. So, the size of the hypothesis space is 3 to the power n.

If we take this particular hypothesis space, the size of the hypothesis space is 3 to the power n and if we plug it in here in the equation that we have then what we get is m greater than equal to 1 by epsilon log of 3 to the power n plus log of 1 by delta and log of 3 to the power n is just n log 3. Now, we can put values for epsilon and delta and given the number of features, we can find out the value of m and these are some of the values that I have worked out. If epsilon and delta are both 0.05 then if you have 10 features then you require 280 examples. If n is 10 epsilon is 0.05 delta is 0.01 then you require 312 examples. If epsilon and delta are both 0.01 then you require 1560 examples. If epsilon and delta are 0.01, n is 50, it requires 5954 examples.

And these values are coming from this formula and this result holds for any algorithm which given a training set can find a consistent hypothesis, and it is up to you to design an algorithm which processes a training set and find out if there is any consistent hypothesis. If there are many consistent hypotheses, your algorithm will come up with one of them. One such algorithm we will just discuses very quickly that can come up with the conjunction of Boolean literals given a training example is called finders and I will quickly go through the algorithm.

(Refer Slide Time: 09:10)



Concept Learning Task

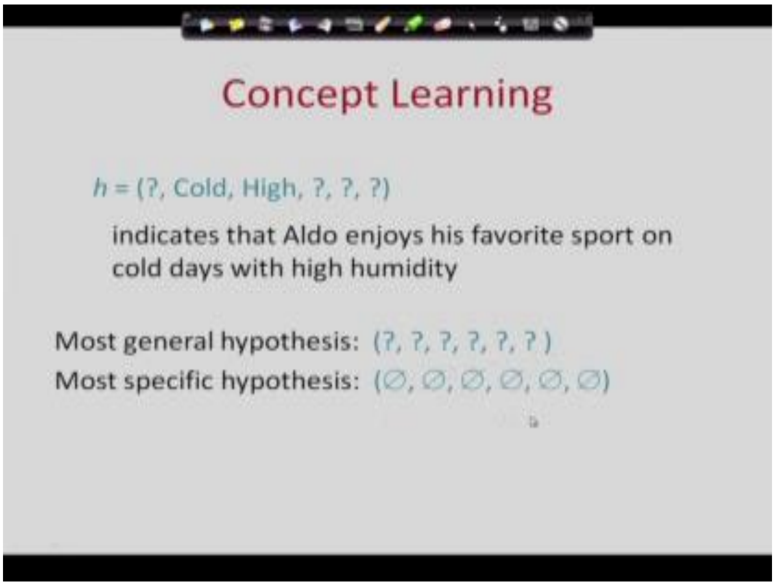
"Days in which Aldo enjoys swimming"

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- Hypothesis Representation: Conjunction of constraints on the 6 instance attributes
  - "?" : any value is acceptable
  - specify a single required value for the attribute
  - "∅" : that no value is acceptable

So, here we have a concept learning task and we have 6 or 7, we have 6 different attributes sky, air, temperature, humidity, wind velocity water temperature forecast enjoys sport. So, enjoys sport is our target variant and these are our input attributes. Now, a conjunction of constrains conjunction of literals is our hypothesis space and a conjunction can be represented by, you see for every attribute in a particular hypothesis that attribute may occur in the positive form, may occur in the negative form or may not occur. If it does not occur any value of that attribute is acceptable. So, I put question mark when any value is acceptable and otherwise I specify whether it is there in the positive from or negative form, 5 means no value is acceptable..

(Refer Slide Time: 10:15)



Concept Learning

h = (?, Cold, High, ?, ?, ?)

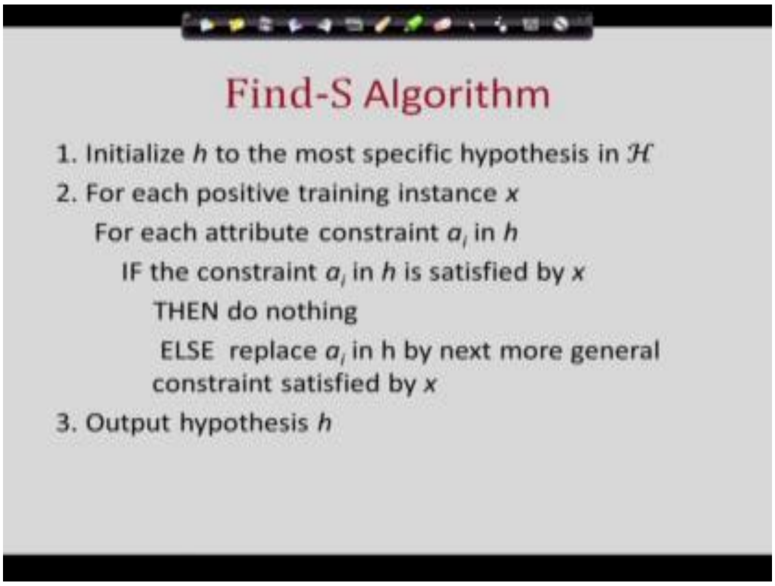
indicates that Aldo enjoys his favorite sport on cold days with high humidity

Most general hypothesis: (?, ?, ?, ?, ?, ?)

Most specific hypothesis: ($\varnothing$, $\varnothing$, $\varnothing$, $\varnothing$, $\varnothing$, $\varnothing$)

So, this is a representation of one such hypothesis, which says the x 1 can be anything, x 2 must be cold, x 3 must be high, x 4, x 5, x 6 can be anything right. So, this is one particular hypothesis. The most general hypothesis is one where any attribute can take any value and the most specific hypothesis is one where no value is possible

(Refer Slide Time: 10:48)

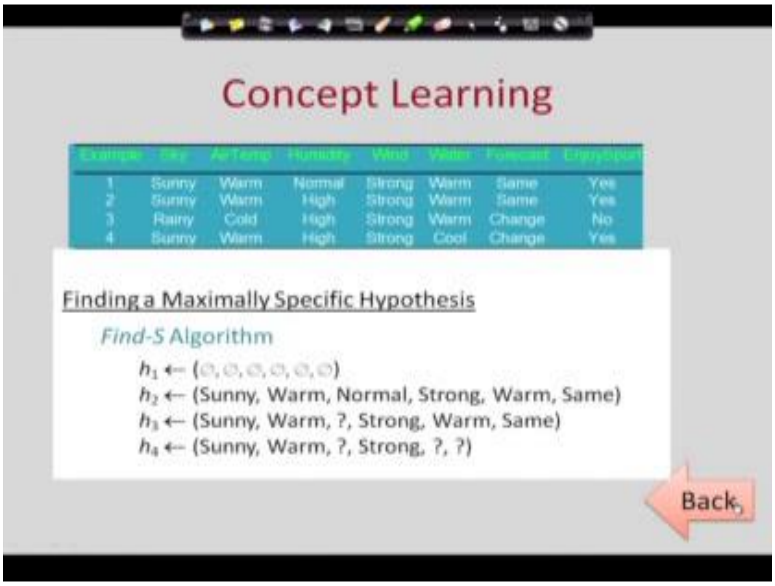


Find-S Algorithm

```
1. Initialize $h$ to the most specific hypothesis in $\mathcal{H}$
2. For each positive training instance $x$
     For each attribute constraint $a_i$ in $h$
        IF the constraint $a_i$ in $h$ is satisfied by $x$
           THEN do nothing
           ELSE replace $a_i$ in h by next more general
           constraint satisfied by $x$
3. Output hypothesis $h$
```

Now, what the find s algorithm does is that it initializes small h, small h is our current hypothesis. It initializes small h to the most specific hypothesis in the hypothesis space in this case 5, 5, 5, 5, 5, 5 then it takes a positive training instance and for each attribute a i in the hypothesis, if the constraint a i in h is satisfied by h will do nothing, otherwise if it is not satisfied we will replace a i by the more general constrain satisfied by x, for example, if a i was given as true and it is not satisfied then.

(Refer Slide Time: 11:46)



Concept Learning

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Finding a Maximally Specific Hypothesis

Find-S Algorithm

$h_1 \leftarrow (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$

$h_2 \leftarrow$ (Sunny, Warm, Normal, Strong, Warm, Same)

$h_3 \leftarrow$ (Sunny, Warm, ?, Strong, Warm, Same)

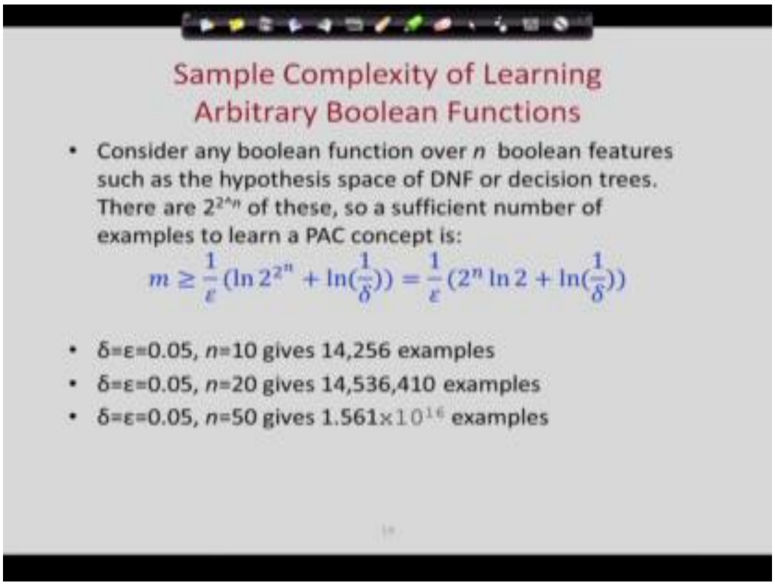
$h_4 \leftarrow$ (Sunny, Warm, ?, Strong, ?, ?)

Finally, after processing all the positive examples, we will output the hypothesis h that we get. So, suppose you are given this training set and you run the algorithm then initially h 1 is the most specific algorithm, most specific hypothesis where all attributes where none of the attributes are permissible.

Then after you see the first example 1, where sky is sunny temperature, warm humidity normal etcetera then h 2 becomes sunny, warm, normal, strong, warm same which agrees with the first training exam, it is the most specific hypothesis that agrees with the first training example, then you bring in second training example and the second training example as humidity is high which is not agreeing with the current hypothesis. So, humidity you set as question mark which is the more general form. Then we look at the next training examples and you find out that water is not agreeing, water and forecast are not agreeing. So, you put these things. So, this is the final hypothesis that you have h 4 is sunny, warm, question mark strong question mark question mark.

Now, this is the find s algorithm which given a set of training examples finds a consistent hypothesis, and if more than one consistent hypothesis exists it will come up with the one which is more specific and according to our theory for this particular hypothesis we can have this relation. The relation states that, if we look at these many numbers of examples then we can produce these guarantees.

(Refer Slide Time: 13:38)



Sample Complexity of Learning Arbitrary Boolean Functions

- Consider any boolean function over *n* boolean features such as the hypothesis space of DNF or decision trees. There are $2^{2^{\wedge}n}$ of these, so a sufficient number of examples to learn a PAC concept is:

$$m \geq \frac{1}{\varepsilon}(\ln 2^{2^n} + \ln(\frac{1}{\delta})) = \frac{1}{\varepsilon}(2^n \ln 2 + \ln(\frac{1}{\delta}))$$

- δ=ε=0.05, *n*=10 gives 14,256 examples
- δ=ε=0.05, *n*=20 gives 14,536,410 examples
- δ=ε=0.05, *n*=50 gives 1.561×10^{16} examples

If your hypothesis space is all possible Boolean function, the size of the hypothesis space if you have n attributes is 2 to the power 2 to the power n which we have already seen in this case, m will be given by this relation, m greater than equal to log of 2 to the power 2 to the power n plus log of 1 by delta into 1 by epsilon and if you put some values of epsilon and delta, for example, epsilon and delta are 0.05, n is 10, you require 14,256 examples, but if n equal to 20 you require 14.5 million examples and n is 50, you require 1.561 into 10 to the power 16 examples. So, if the number of attribute is large, the numbers of examples required is exponentially which makes this problem infeasible. With this we stop today.

Thank you.