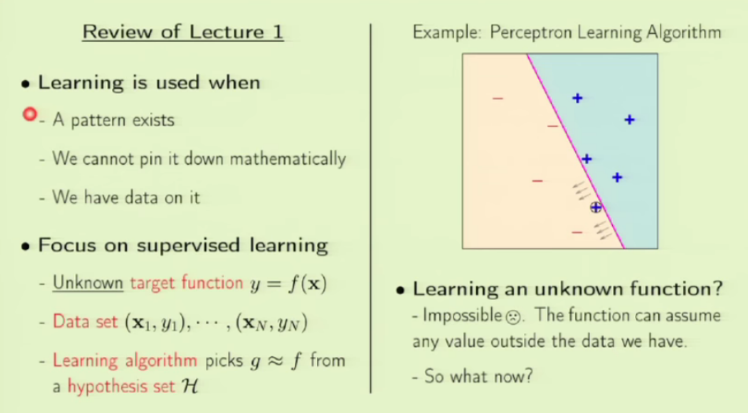
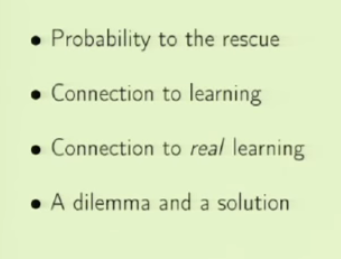
Lecture 2:

If there is no pattern, learning fails. We can still try for the purpose of knowing whether there is pattern in the data.

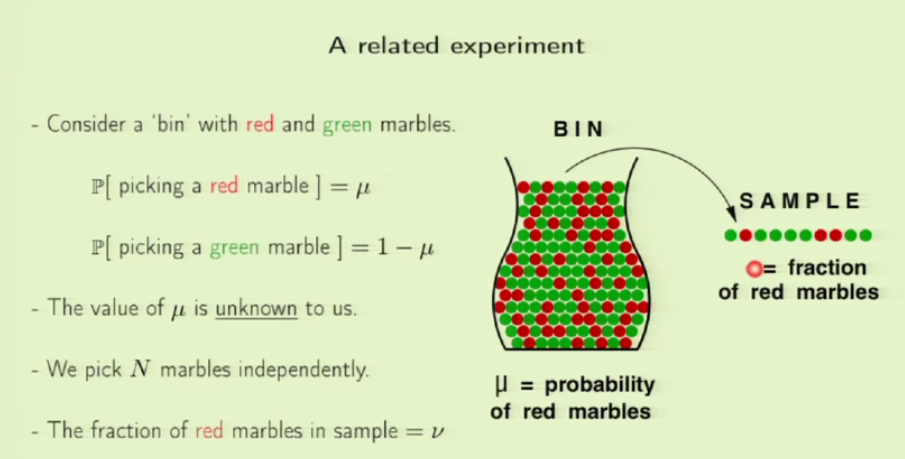
If we can pin it down with math, we can still use ML.

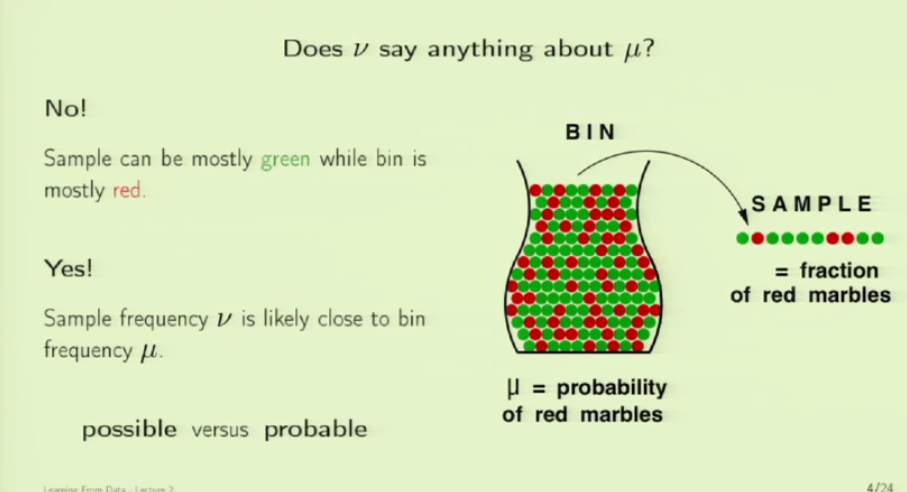
Without data? Forget it, nothing’s possible.

The main property of target function is that it is unknown. Since the data contains the output labels or classes, we call it supervised learning.

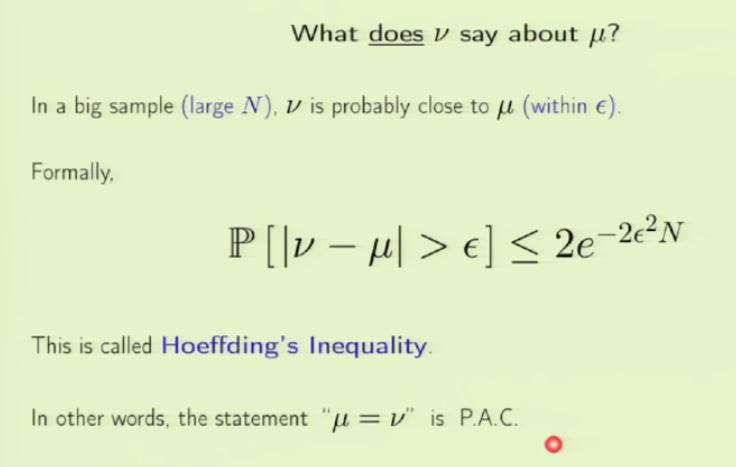
**Is learning feasible?**

We will start with a probabilistic situation. Can we capture something outside the data?Then translate it to 2 stages – connection to learning and real learning. There is a dilemma with this which will be discussed.

We have a bin with red and green marbles. Experiment is to pick a marble. This is just a representation of an experiment with binary outcome. We have no idea about what u is. Different samples will contain different proportions of red and green marbles. V is fraction of red marbles in sample.

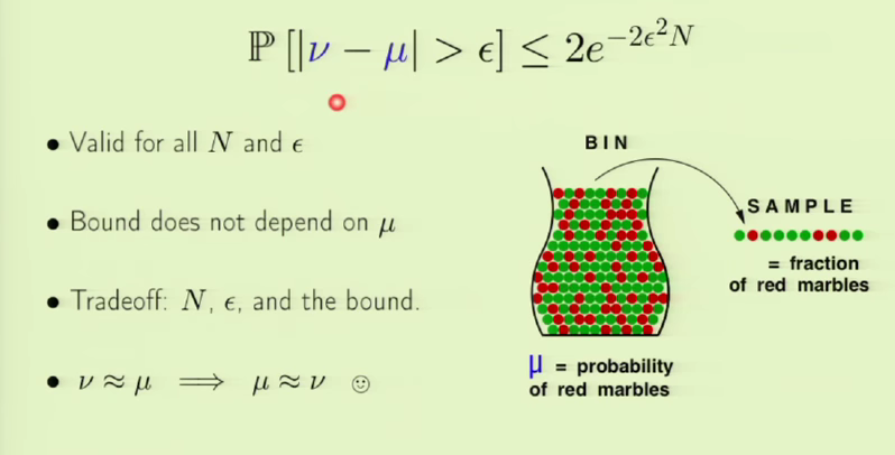


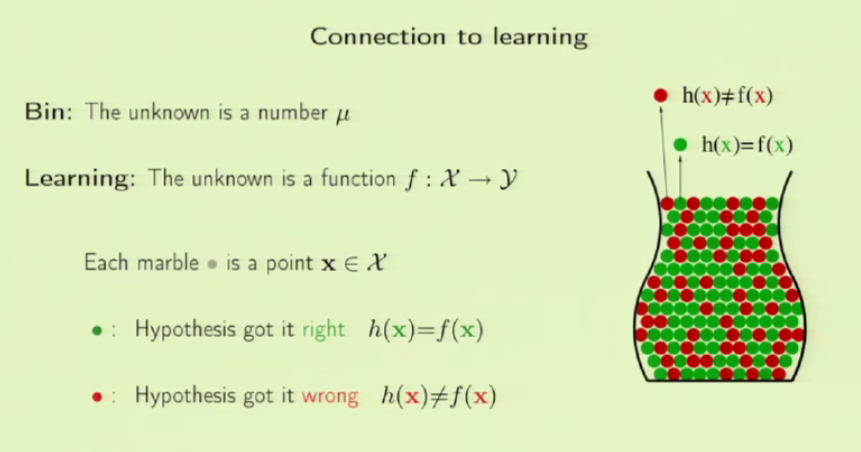
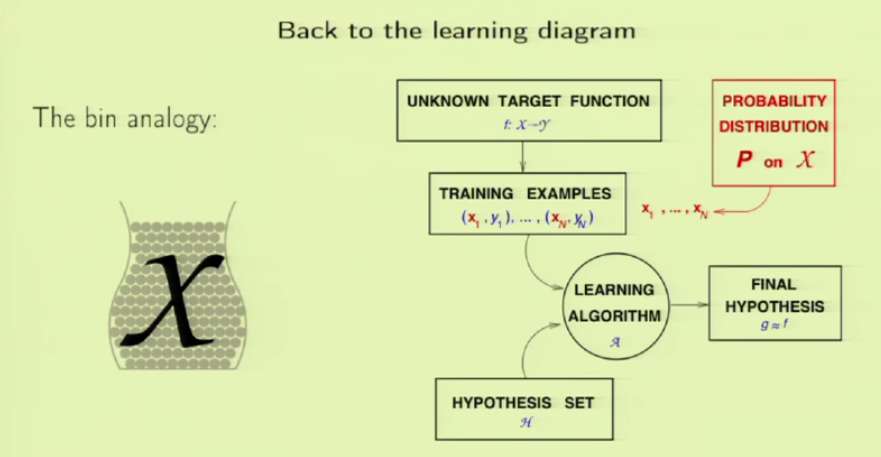
Does v say anything about u? A YES and a NO. Consider the example of a poll. There are million voters, you take sample of 3000. How will they vote? Possible vs Probable. Absolutely certain vs almost certain.

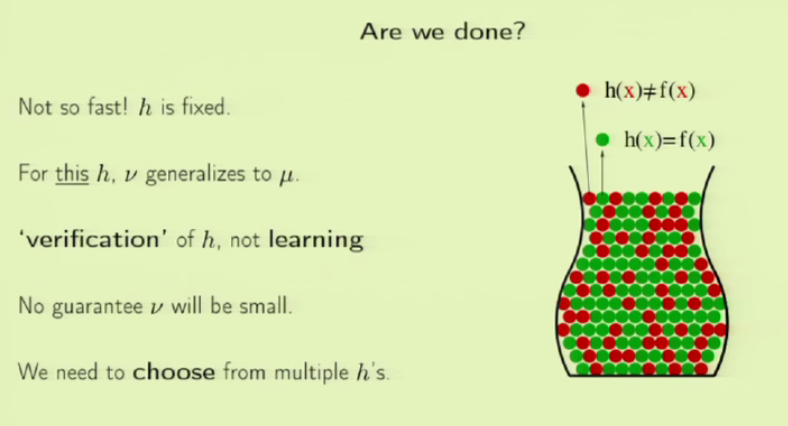
If we are claiming something to be of low probability, then it must be a bad quality. We want v – u to be within E.The bad thing in this case is for E to be > v – u.

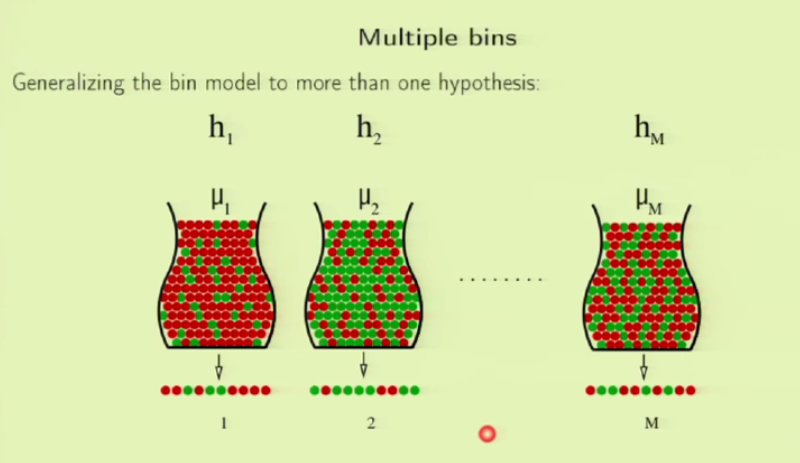
Probably approximately correct.

Law of large numbers.

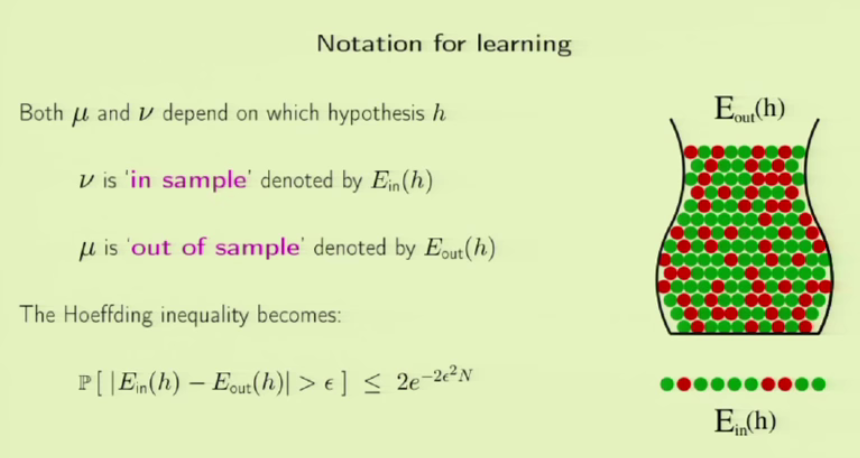
The right hand side doesn’t have u, which is a good advantage. The smaller the E value is the bigger the N you need to compensate the E value to maintain the same level of probability bound. When you run the experiment, v is the random component, where as u though unknown is the happy constant.From the equation, you are trying to infer u from v. But that’s not the actual cause and effect, its u which effects v, not the other way. But still we use it that way. Its fine since the form of probability is symmetric.

The bin problem is simple, but in real, a learning problem will be complex. Relating the bin problem to credit card proble, let each marble be a sample. Green marbles is Hypothesis getting it right. If H matches with actual output, mark them green. The wrong H are marked in red. Now we have a new component added to learning problem. In addition to data, we now have the probability linked to it. P can be anything, no restrictions on it. The p choice will effect whether its green or red marble. But still P can be anything. Now we use P to generate the data points, each being independent.

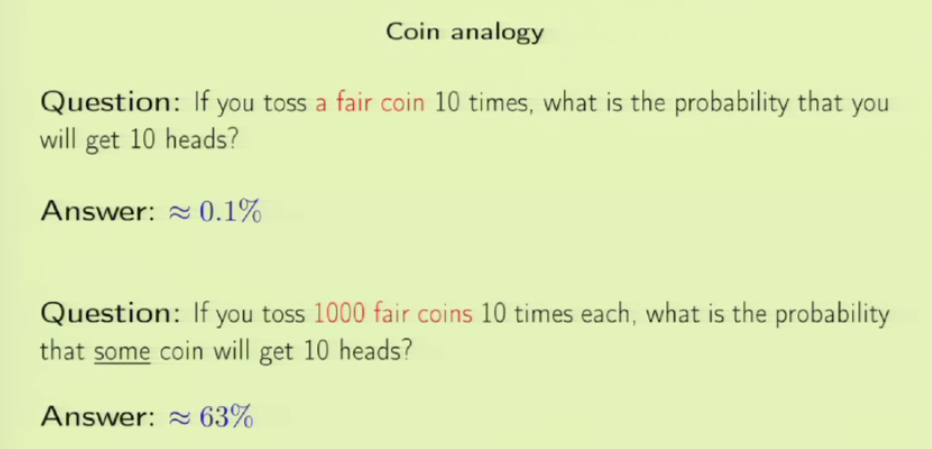
We have a problem here. We should choose H from multiple hs. We will have multiple bins each with its own h1 and u1. We will choose the best bin and then its h and u values.

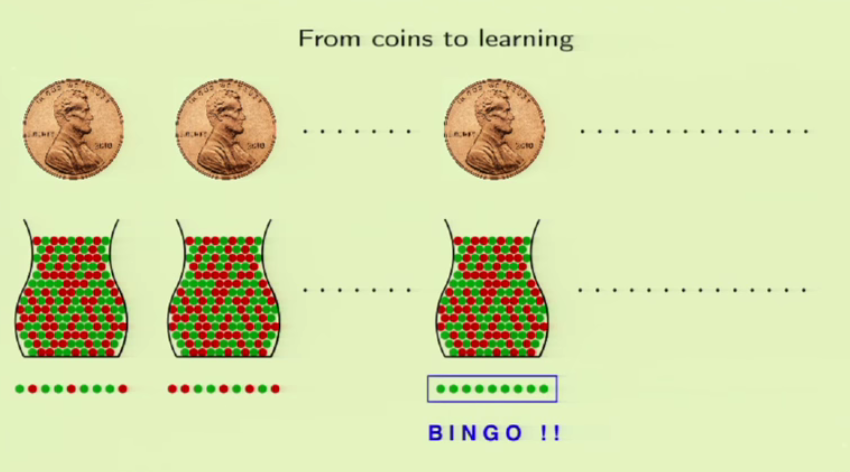
We can think of it this way: we set a weight vector W for a specific H, and then decide if it’s a green or red marble. For another H , we use different W and come with different outcome.

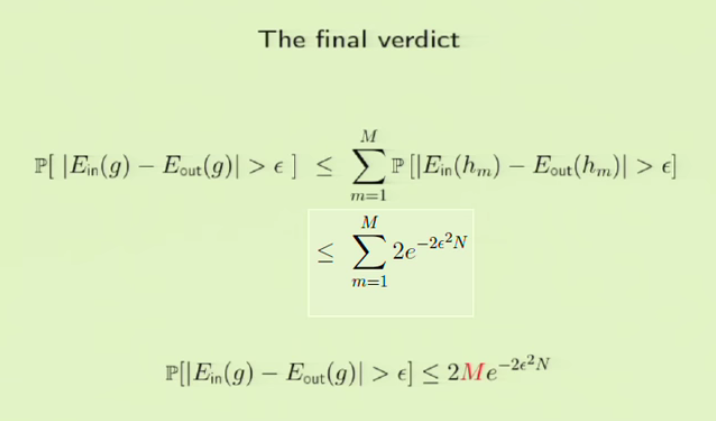
Each h has different u and v proportions.

using new and relevant notations for u and v.

Are we done now? Not yet. Hoeffding doesn’t apply to multiple bins!!! What can be done now? Why doesn’t it apply? Take a coin and flip it 5 times and record the output. Will anyone get all 5 heads? If yes, it’s a biased coin?

We see that when coin is tossed more and more, those few flips which will give certain higher probabilities, being disjoint, club together to give a better probability as we got with 1000 coins flipped 10 times each.

Is it really BINGO? No. the hypothesis which we think approaches the target value might get terribly diluted for new data.



This gives the outer bound or the worst case.

As the model gets complex(M gets bigger), it generalizes less and memorizes more as discussed in Q & A.

**Q & A:**

Hows v = u, u = v? Given probability, you try to take samples. In statistics it’s the reverse. You are already given the sample and trying to infer its probability. In ML, v is the one that keeps changing for some constant u. Each h is a function. Complexity of individual h vs collective h.

G is just the final H we choose.

How to use probability equation for multiple classes output? Can be done with some technical changes to notation.

Why multiple bins? With single bin we pick a h, draw some conclusions from it relating to the data. But we need multiple h for getting at better results. So we need multiple bins so that we can play around with different h until we reach at an efficient h.

We set probability on the data alone but not on H.

What if we have multiple hyperplanes that separate our linear data? We have some certain metrics which will help us in deciding which model. This includes even for multiple models with zero error.

G is the final h. Given data, the ml algo goes through the data and magically comes up with one final h. In the process of doing this, it actually tries out different h. By changing some parameters or weights, we have different h. we do this until the h generalizes for data.

Verification – we predict some output for sample, then verify how close it is to the output of the data or bin. Verification is feasible. When we do it for many samples, M samples, we use learning.

The bin with gray marbles is the general data or population. As we pick h, we get colored marbles.