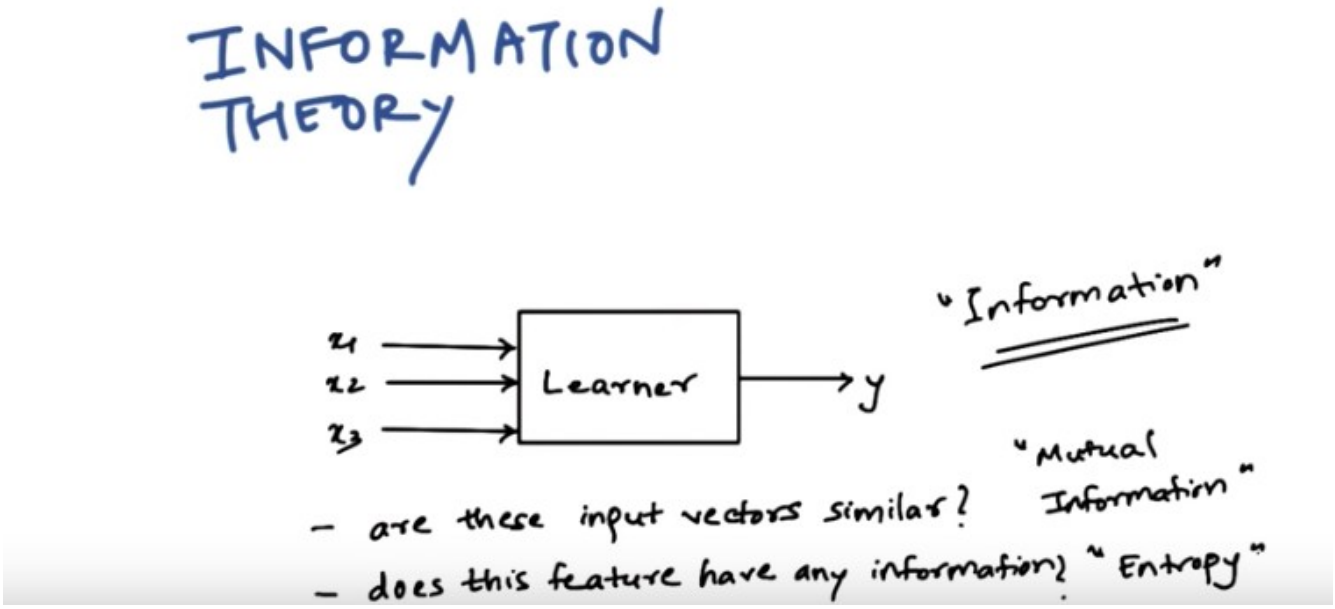
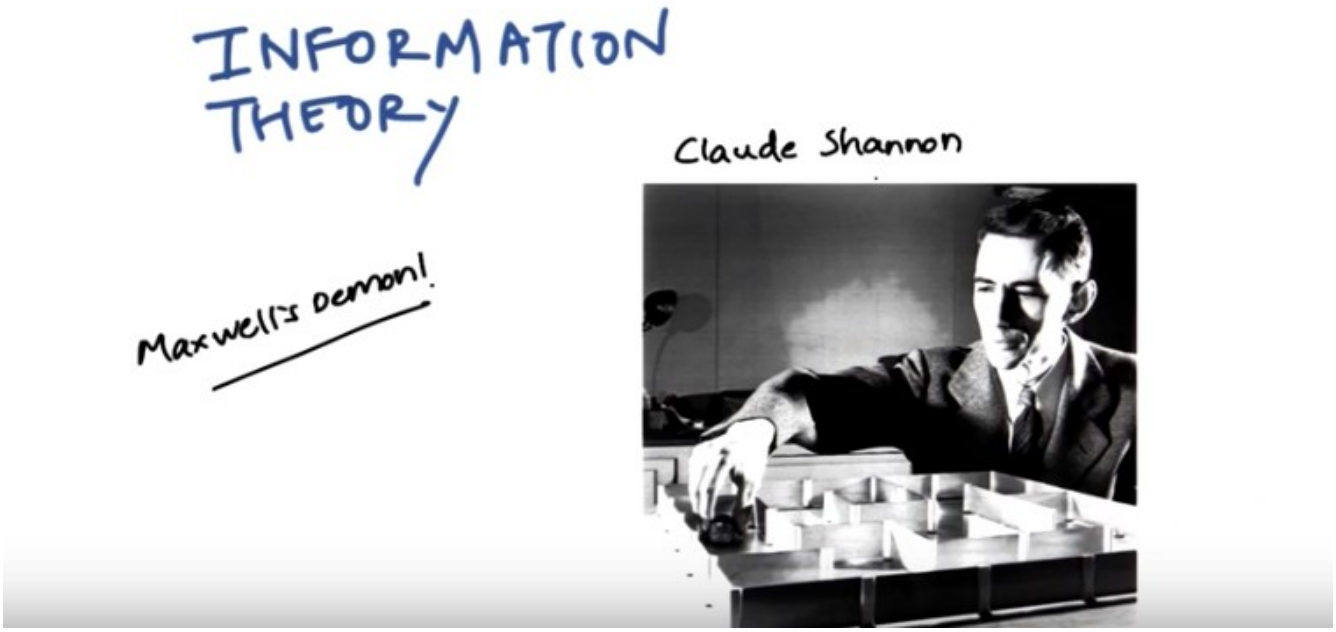
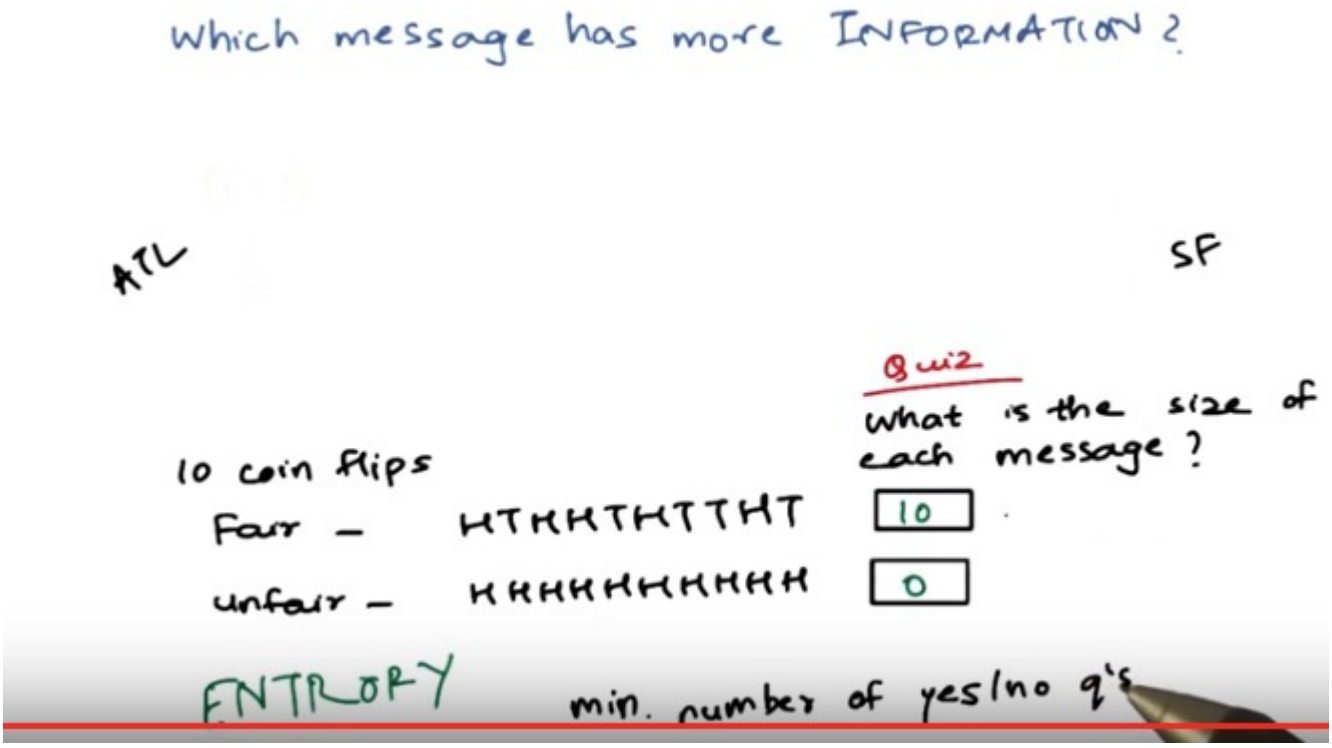


1. Hi, my name is Bushger, and today we are going to talk about information theory. Now, information theory is not really a machine learning algorithm, so we'll, kind of, first understand why we need to learn information theory. Usually, you could teach a whole course on information theory, but for now, it is sufficient to know the basics. So first we'll try to understand where information theory is used in motion learning. So consider this to be any machine learning algorithm. For example, let this learner be, or a decision learner, now we have several inputs, x1, x2, x3, and one output. For simplification, let's assume that this is a regression problem. That's why we have one output. We want to ask interesting questions like how is x1 related to y, x2 related to y, x3 related to y. Why do you want to ask such questions? If you remember, from our IDT algorithm, the first step is to find out which input best splits our output. So we need to find out which of these, x1, x2, or x3 gives you the most information about y. So we have to first understand what the word information in information theory means. In general every input vector and output vector, in the machine learning context, can be considered as a probability density function. So, information theory is a mathematical framework which allows us to compare these density functions, so that we can ask interesting questions like are these input vectors similar? If they're not similar, then how different they are? And so on. We call this measure as mutual information. Or we could ask if this feature has any information at all. So we'll call this measure entropy. So we are going to find out what these terms mean, how they're related to information learning in general, and we'll briefly look at the history of this field.



2. Information theory has a very interesting history. Claude Shannon was a genius mathematician who was working at Bell Labs who came out with this information theory. He's also called as the father of the information age. Why it is interesting, is because at the time Bell Labs had a communication mechanism set up and they had just figured out long distance communication, but they had no idea how to charge people. So you could send a message and they would charge you per message, or they could find out how many words were in those message and they would charge you per word, but none of them made sense because you could sometimes write shorter sentences and can be much more information. So it really became necessary to find out what is information and Shannon was the first person to ever try to work on that problem and figure something out. But information theory has also a background from physics and that is why it has words like entropy in it. The physicists who studied thermodynamics were the first scientist to actually understood information. If you are interested in learning more about the physicists background of information theory, you should read up on Maxwell's Demon. Maxwell's Demon is a very famous part experiment that Maxwell came out with. In physics we believe that energy can neither be created or destroyed, but Maxwell's Demon proves that energy can, can ordered into information and the combination of energy and information can neither be created or destroyed. But let's come back to the big world and let's discuss how Claude Shannon looked at it. So his task was to send messages from one place to the other and try to figure out which message has more information. So, let's start with a simple example.

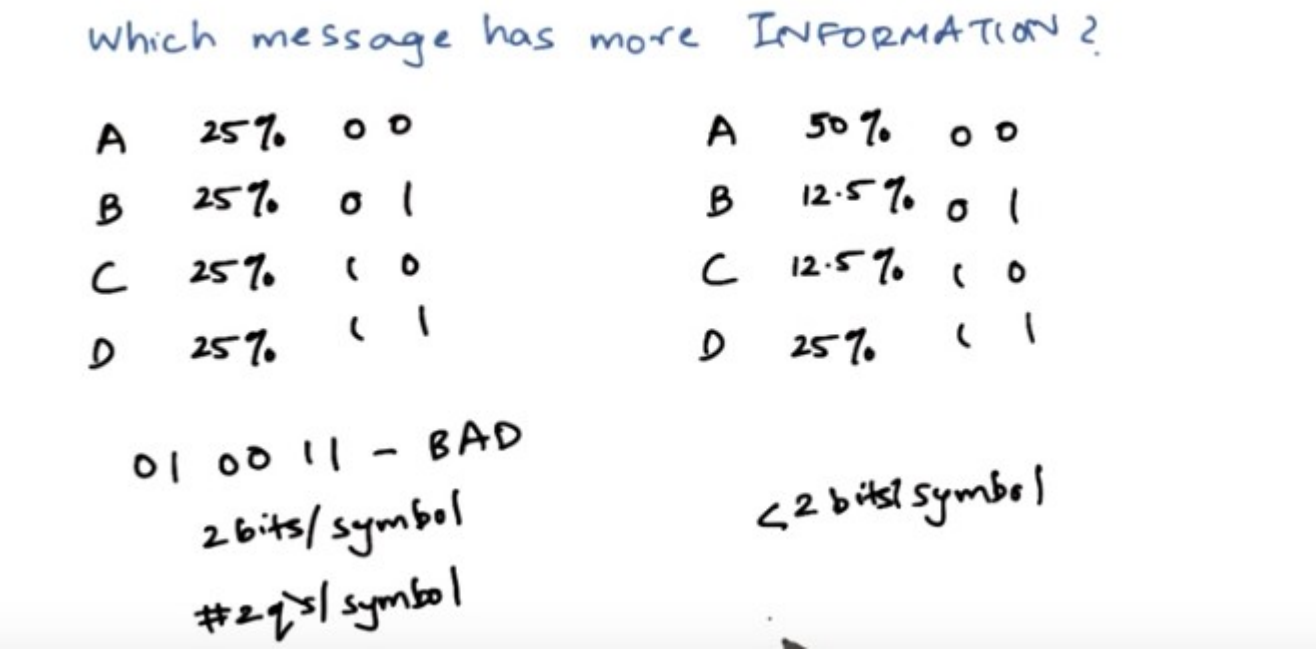


(由後面知: Tao: number of bits per symbol = information 的多少 = entropy = randomness)

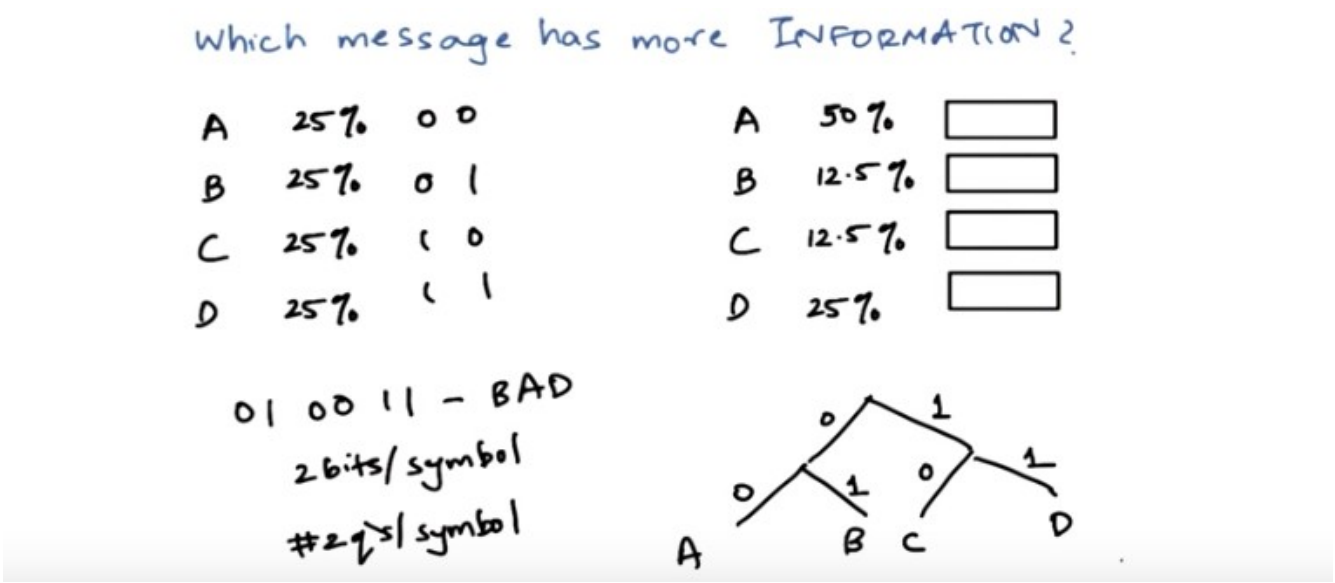
3. Let's assume that you want to send a message from Atlanta to San Francisco. And to make it easier, let's assume that we want to send a simple message which consists of n coin flips, or the output of ten coin flips. Let us construct two messages out of coin flips. Now, I have two coins, but you see these coins are different because this one has a heads and a tails, it has two different sides, And this one has both the sides which are very similar looking. So its a biased coin so every time I flip its going have the same state. While when I flip this it might either end up here, 50% of the time or end up here 50% of the time. So we'll construct two messages after flipping both of them and recording what their state is. So here it is, I did ten coin flips with the fair coin I got a few heads, a few tails, in this particular sequence. The unfair coin, I'm calling every state as a head state and I basically saw ten heads. All right? If you also observe the fair coin I have like five heads and five tails. So the probability, so it, so it wa, so it is a fair coin. I got five heads and five tails, so it is a fair coin. So, if I had to transmit this sequence, how many bits of message will I require? Let's assume that I can represent this sequence using ten binary digits. A zero representing heads, one representing tails and I can write down this sequence as zeros and ones using ten bits. I can also write down the same sequence with the, of the unfair coin using those ten binary digits. So I'll get something like zero, one, zero, zero, one, zero, one. And here, everything will be zeros. Let's assume I have to transmit these two particular sequences from Atlanta to San Francisco. What will be the size of each message in case of the fair coin and the unfair coin? What do you think? Write down your answers here.

4. So, in the first case, we will need ten bits for each of those ten flips. But in the second case, we don't need any bits because the result of the flip is always going to be the same. You don't even have to send this message. The folks at San Francisco will already know what is the result of those ten flips. So realize what we've discovered here, if the output of this coin is predictable, you don't need to communicate anything. But if the output is random, you need the communicate the result of each and

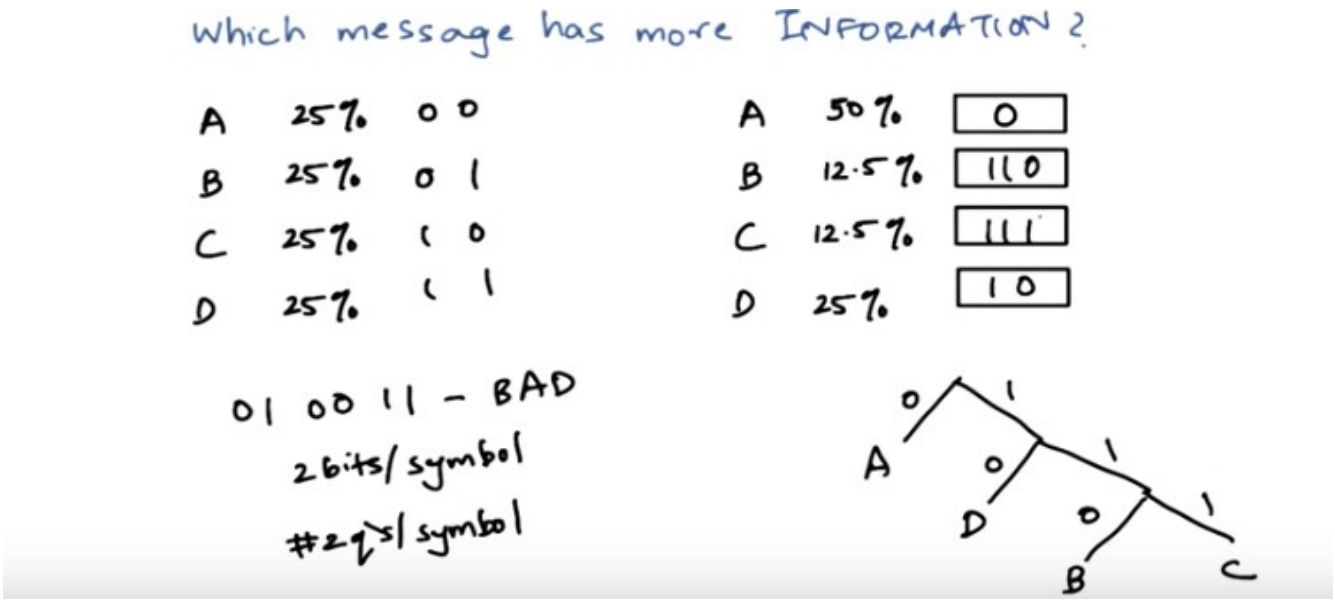
every flip. So more information has to be translated. If the sequence is predictable or it has less uncertainty, then it has less information. Shannon described this measure as entropy. He said, if you had to predict the next symbol in a sequence, what is the minimum number of yes or no questions you would expect to ask. In the first example, you have to ask a yes or no question for every coin flip. So you have to ask at least one question for every flip. In the unfair coin, you don't have to ask any questions. So the information in the second case is zero, while the information in the first case is one. Let's consider another example to understand this better.



5. Let's consider that we want to transmit a message which is made up of four words, A, B, C and D. And let's assume that all the four letters are used equally in the language. They occur, the frequency of each letter occurring in the language is equal. So, you can be present A, B, C and D in binary, with two bits each, like so. Which means if we have a sequence such as zero, one, zero, zero, one, one, the six bits spell out the word BAD, bad. So basically we'd require two bits per symbol. Other way to look at this sequence is that you need to ask two questions to this sequence to at least recognize one symbol. So, two bits per symbol also means that you have to ask two yes or no questions per symbol. Now let's consider the second message to be made up of the same symbols but in this case A occurs more frequently than B, C or D. D of course more frequently than B and C. and, let's assume that these are the probabilities by which we can see A, B, C, or D. Now, we can do the same thing again, we can use the same binary representation to represent A, B, C, and D. So again, we'll end up with two bits per symbol. But can we do any better? Well, A occurs more frequently than the others so can we somehow use this to our benefit and use a different bit representation to get slightly less than two bits per symbol? Think about it. Can you think of a new representation that might be better?

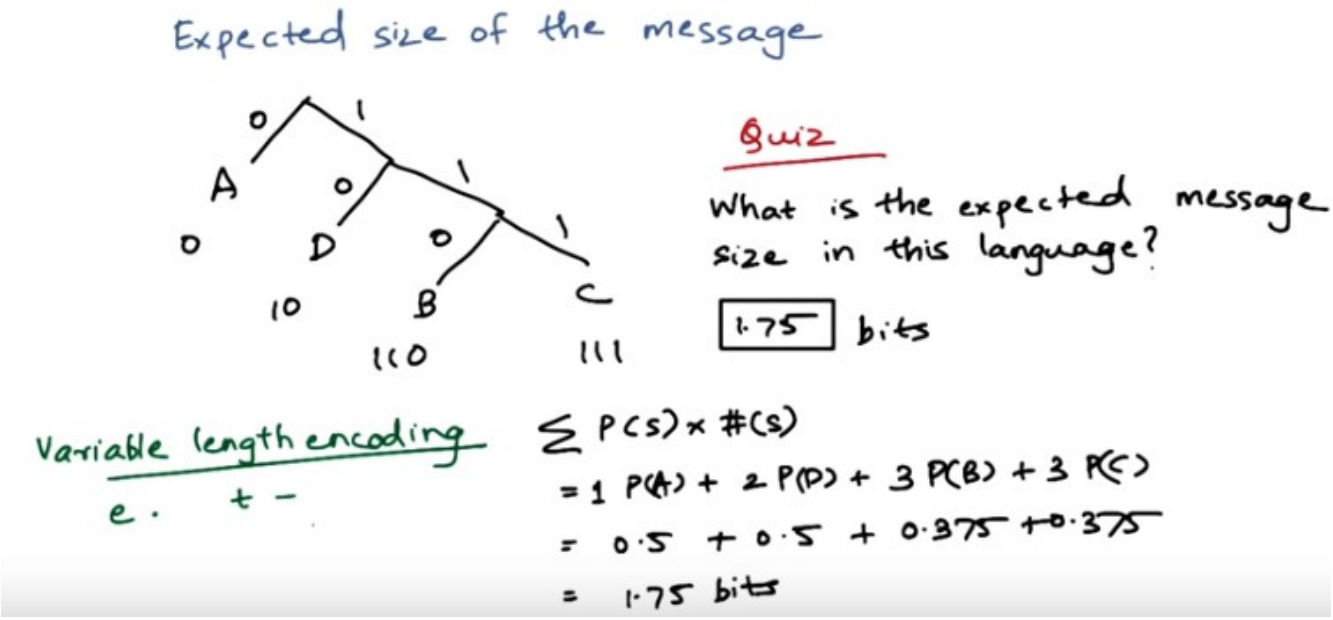


6. Okay, so the way you can think about this question is by looking at the first message and why it makes sense to have two bits per symbol. So you can be present this bit pattern in a tree. So when a new bit comes in, it can be either 0 or a 1. If it's a 0, the next symbol can be another 0 or it can be 1. The same case here, so if there are two 0s, it is an A, if it is a 0 followed by 1, it is a B, if it is a 1 followed by 0, it is a C, if it is a 1 followed by 1, it is a D. So that is why you need to ask two questions to reach either of these symbols.



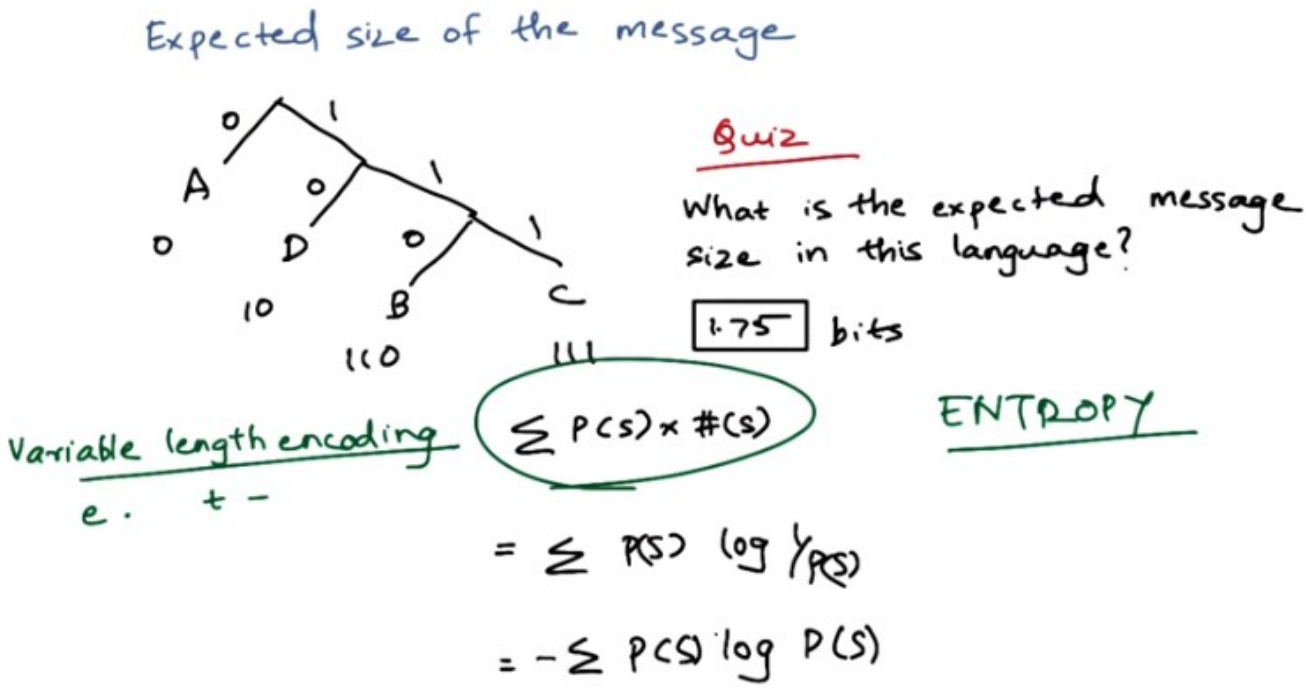
What happens in this new language? So in the this new language, it occurs 50% of the time. So we can directly ask if it is A or not A. So that's represent that has 0 or 1, and let A be 0. So now we got our A as just a 0. Now if we go on the one branch we can again ask, if it was a 0 or a 1. Now observe that D

occurs twice as frequently as B or C. So, in this case we can be present D here using 1, 0 and then B or C can occur on this branch but both cannot occur at the same place, so we need to differentiate between them using another symbol. So let's do that using 0 and 1 again. So this can be B and this can be C. So B is basically 1, 1, 0 and C will be 1, 1, 1. Now have we actually saved any bits per symbol, have we saved the number of questions we asked? Yes because A occurs more frequently and I needed to ask only one question. 下圖的問題: But what is the exact number of questions we are to ask for symbols?



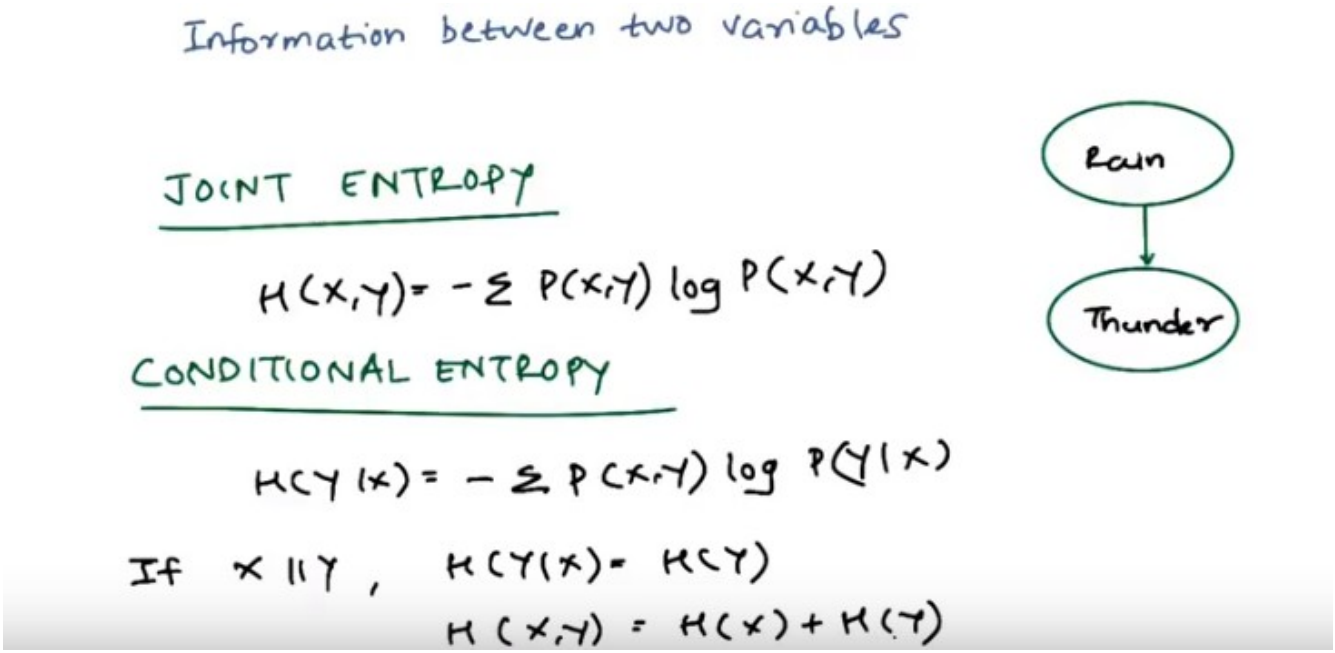
7. Now let's convert this into a quiz. Now remember, that we use this tree to find out our representation. We need one bit for sending A, two for D, and three each for B and C. So, if you send a message in this language, what is the expected message size? Write your answer down here. Remember the unit for this answer is going to be in bits. Go.

8. Okay, so will you work this out? You will need to know the frequency of A, B, C, and D. But we already know that. You will also need to know how many bits each of those symbols require. Now we also know that. So, we will calculate the expected number of bits to transmit each symbol and then add them up. So for any symbol, the expected number of bits is given by the probability of seeing that symbol, and the size required to transmit that symbol. And we add them up for all the symbols in the language. This is going to give us 1.75 bits on an average. Now, since we had to ask less questions in this language, than the previous language, this language has less information. This is also called as variable length encoding. This should give you some idea into figuring out why some symbols in Morse code are smaller than others. In the English alphabet, the letters e and t occur most frequently. That's why, in the Morse code, e is generated by a dot and t is generated by a dash. Since e and t occur more frequently, they have the smallest message size.



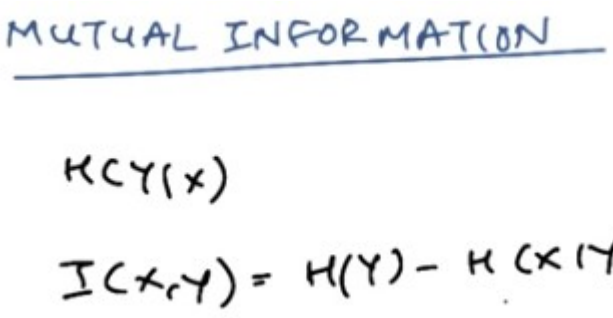
Tao: number of bits per symbol = information 的多少 = entropy = randomness

This measure, which calculates the number of bits per symbol, is also called as entropy. And it is mathematically given as this formula. To make it more legible, we need to find out how to denote the size of s more properly. The size of s is also given by the log of 1 upon the probability of that symbol. So the formula of entropy is given as this.



最後兩式不要去證, 記住就是, 要證應該也是 obvious, 只是不想花時間.

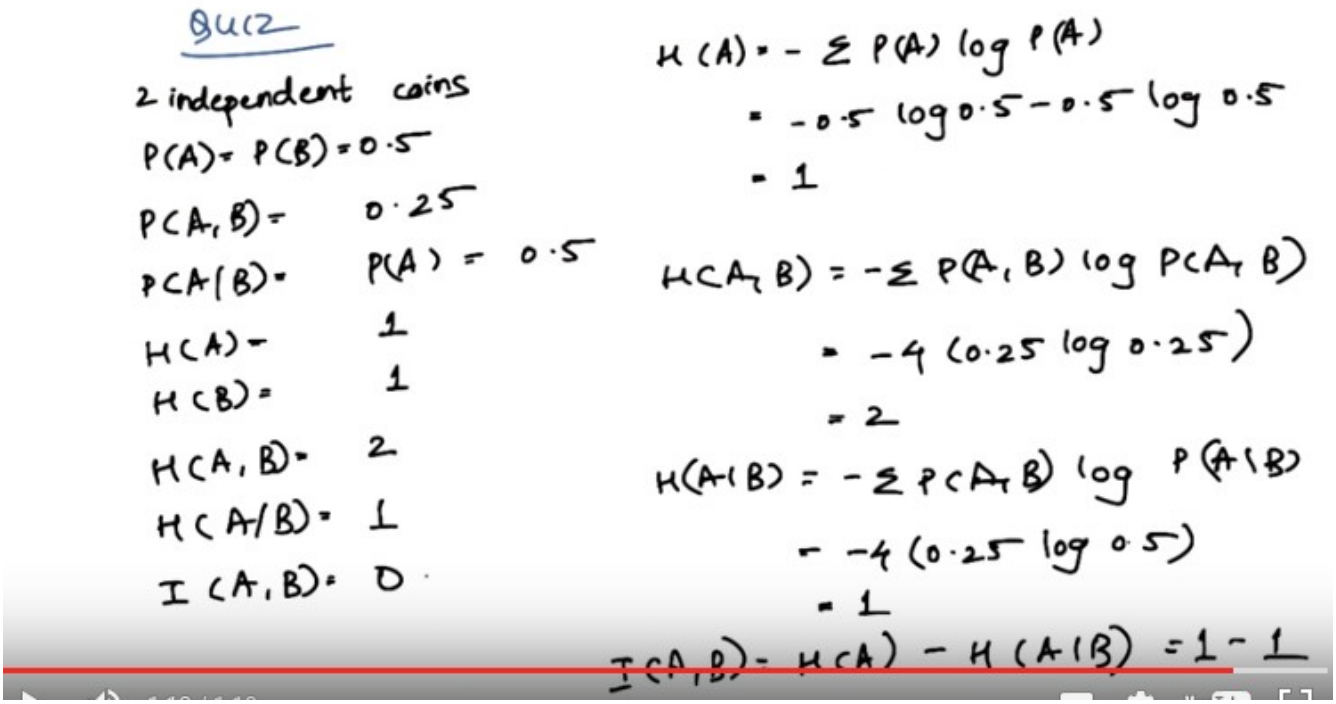
9. Now we know what is the information in one random variable. Now assume that I told you to predict if you're going to hear thunder or not. Well, that's very difficult. But what if I tell you if it is raining or not. Your guess regarding the thunder is going to be significantly better. So there is some information in this variable that tells you something about this variable. We can measure that in two different ways. The first one is called as joint entropy. Joint entropy is the randomness contained in two variables together as given by H of X comma Y. And, as you can predict, it is given by this particular formula, which is the joint probability distribution between X and Y. And, as you predicted, it is given by this particular formula where P of X comma Y is the joint probability of X and Y. The other measure is called as conditional entropy. Conditional entropy is a measure of the randomness of one variable given the other variable. And it is generated by H of Y given X. Now, to understand these two concepts, you have to imagine what happens when X and Y are independent. If X and Y are independent, then the conditional probability of Y given X is just the conditional probability of Y. It's quite obvious, right. If two variables are independent of each other, Y variable doesn't get any information from X at all. The joint entropy between X and Y, if X and Y are independent, is the sum of information of both X and Y. That is why the entropies have been added here.



From Instructor Notes: 上式寫錯了, 正確的是 I(X, Y) = H(Y) - H(Y | X)

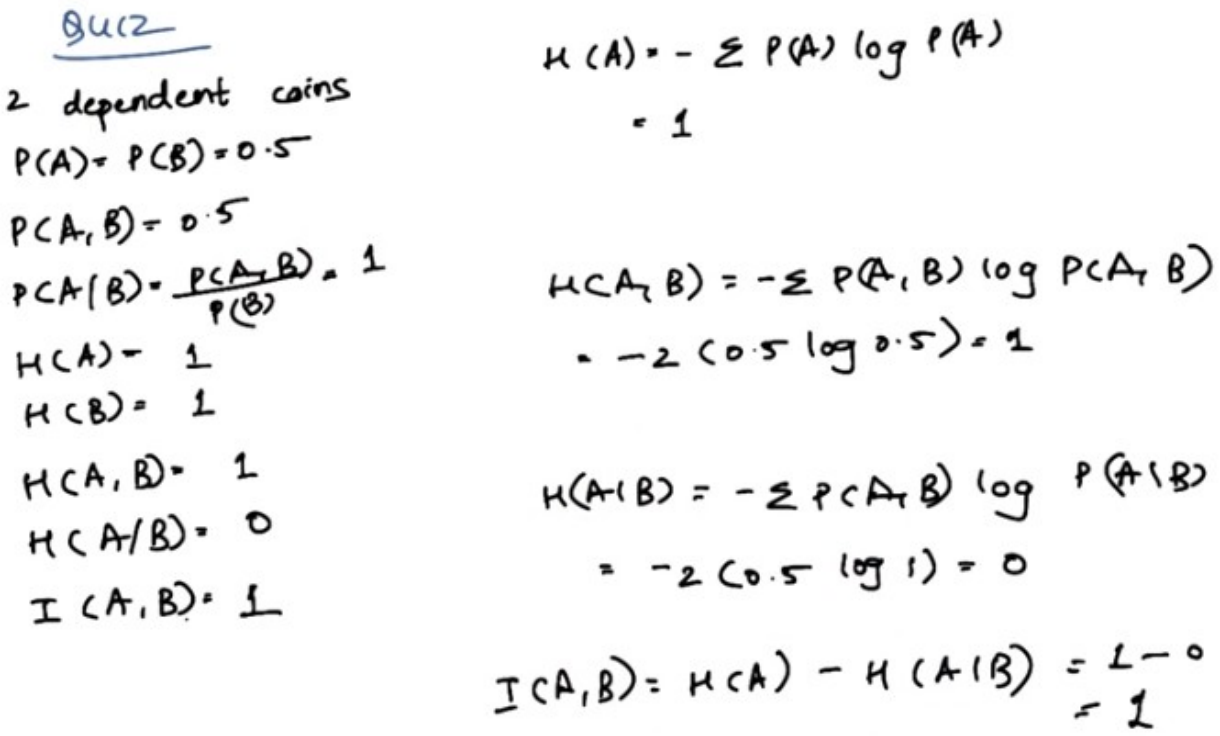
10. Although conditional entropy can tell us when two variables are completely independent, it is not an adequate measure of dependence. Now consider the conditional entropy of y given the variable x. This conditional entropy may be small if x tells us a great deal about y or H(y) is very small to begin with. So we need another measure of dependence to measure the relationship between x and y and we call that as mutual information. It is denoted by the symbol I. And it is given as, the entropy of y, subtracted by the entropy of x given y. So mutual information is a measure of the reduction of randomness of a variable given knowledge of some other variable. If you like to understand the derivations for these particular identities, I'll refer you to Charles's notes on, on this topic. But we'll jump directly into an example and try to calculate these values and understand what it means to have a high value of mutual information or low value of mutual information. So let's do that as a quiz.

11. Okay, so this is the quiz. Assume that you have two coins, A and B. Both are fair coins, and you flip both A and B. We will assume in this case that A gives us no information about B, which is how it works in the real world. So the probability of A and B is 0.5. Try to find out the joint probability of AB, conditional probability of A given B. The entropy of A, B, and the joined entropy, the condition entropy and imaging information. 'Kay, refer to the nodes or formulas from previous videos and try to answer these questions.



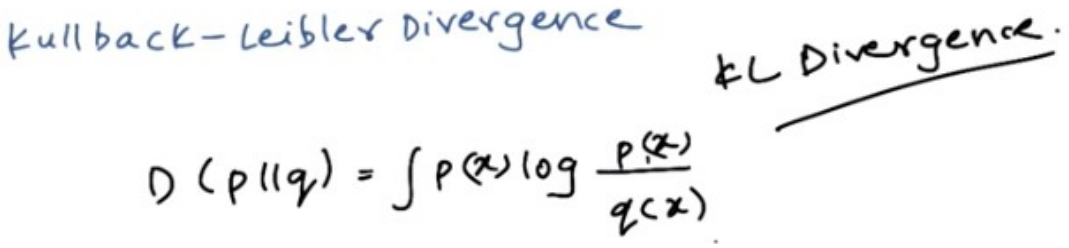
由題目中的 log 0.5 最後得出 1 知, 此處的 log 的底為 2

12. Okay. Let's just simply try and substitute our values in the formulas that we know of. Since A and B are independent events, the triangle probably is given by the product of A, product of probability of A and B, so that gives us 0.25. Probability of A given B, since A and B are [UNKNOWN] of each other. Probability of A given B is just probability of A, which is 0.5. So then the entropy of A is given by this formula. And if we expand on this we get, we get the entropy of A as 1. Similarly the entropy of B is also 1. What is the joint entropy? The joint entropy is given as this formula. So if we substitute the values we get the joint entropy of A and B as 2. What is the condition entropy of A given B? It is given as this formula. And if you substitute the values, we get the conditional entropy as 1. Mutual information between A and B is given by this formula. And if we substitute the values of the variables that we have already calculated, entropy of A and entropy of A given B, we get 1 minus 1, which is zero. So, since the two coins are independent, there is no mutual information between them.

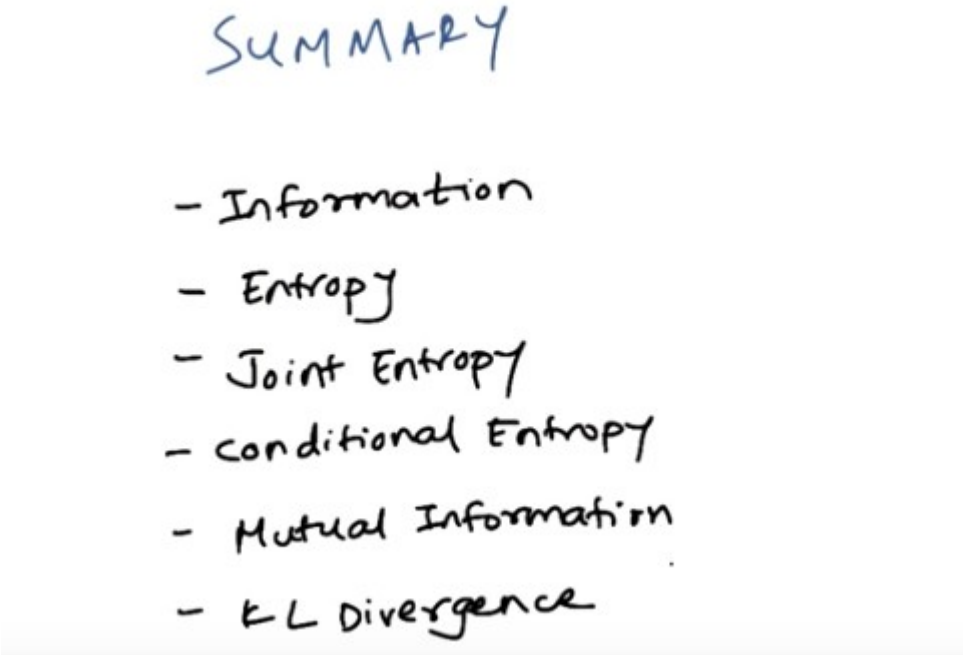


13. Let's do another quiz, where the two coins are dependent on each other. So let's assume a case where you flip two coins, A and B and there's some gravitational force or some kind of weird force acting between them. So, whatever the A flips as, if the A flips as heads, B also turns out to be head. And if A flips as tails, B also comes out to be tails. So complete information is transferred from A and B, and so they're completely dependent on each other. So, find out similarly, what is the joint probability between A, B, conditional probability, their entropies and the conditional entropies and their mutual information. Go.

14. Okay. So, lets start with the joint probability. Now since A and B are both dependent on each other, there are only two possibilities. Both can be heads or both can be tails. So the joint probability is also 0.5. What is the, what is the conditional probability? Conditional probability is given as probability of A comma B upon probability of B. So, conditional probability is 1. Now, what is the entropy of A? It is similar to the last example, because we are still using fair coins. So, the entropy of A is 1. Entropy of B is also 1. What is the joint entropy between A and B? Okay. Let's use this formula, and see if our answer changes. Okay, the joint entropy comes out to be 1, which is different than our last example. What is the conditional entropy? The conditional entropy is given by this formula. Let's substitute the values to find out what we get. Okay, the conditional entropy comes out to be 0. What is a mutual information between A and B, then subtract the entropy of A, and the conditional entropy of A given 0, which is 1 minus 0, which is 1. So the, so the mutual information in this case is 1, while in the previous case, it was 0. So since these coins are dependent on each other, the random variable A gives us some information about the random variable B. This tells you, how mutual information works.



15. To conclude our discussion of information theory, we will also discuss something called Kullback-Leibler divergence. It is also famously called the KL divergence. And you must have heard this term in our previous lectures. So it is useful to realize that mutual information is also a particular case of KL divergence. So KL divergence actually measures the difference between any two distributions. It is used as a distance measure. For this particular lesson, it is sufficient to understand how KL divergence is used to measure the distance between two distributions. The KL divergence is given by this particular formula. And it is always non-negative and zero only when P is equal to Q. When P is equal to Q, the log of 1 is zero, and that's why the distance is zero. Otherwise, it is always some non-negative quantity. So it serves as a distance measure. But it is not completely a distance measure because it doesn't follow the triangle law. But then you should ask yourself why you need to know KL divergence, or where it is used. Usually, and usually in supervised learning you are always trying to model our data to a particular distribution. So in that case our distrib, one of our distributions can be of unknown distribution. And we can denote that as P(x). And then can sample our data set to find out Q(x). While doing that, we can use KL divergence as a substitute to the least square formula that we used for fitting. So it's just a different way of trying to fit your data to your existing model. And we'll come back to KL divergence in some of our problem sets.



16. So, let's summarize what we have learned now. So we, we first understood what is information and we found out that information can be measured in some way and we meausred it in terms of entropy. Then we started to understand how we can measure the information between two way variables. And there we defined terms as, terms like joint entropy, conditional entropy and mutual information. And

then finally we introduced ourselves to a term called a KL divergence, which is very famously used as a distance measured between two distributions. So this is just a primer to information theory and it forms as a base to what is required for you to go through this machine learning course. If you want to learn more about information theory, follow the links in, in the Comments sections. And yeah. Thank you.