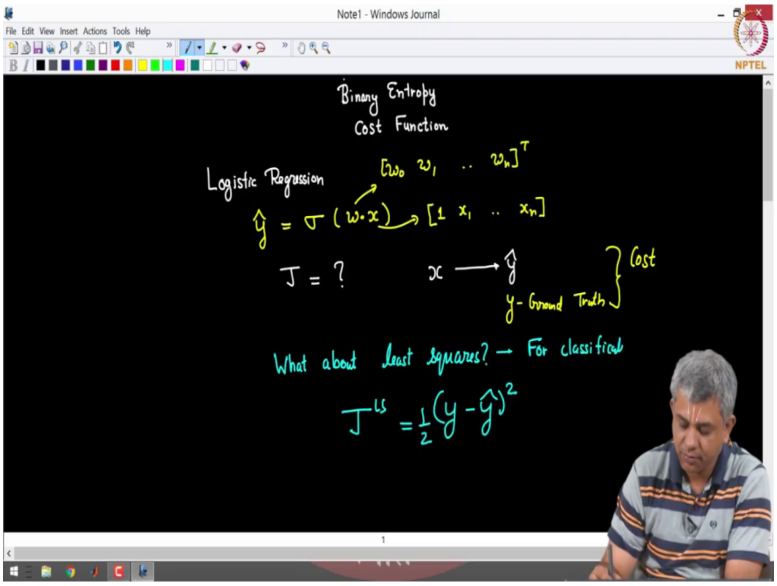
Machine Learning for Engineering and Science Applications

Professor Dr Balaji Srinivasan
Indian Institute of Technology, Madras
Department of Mechanical Engineering

Binary Entropy Cost Function

(Refer Slide Time: 0:28)



Binary Entropy Cost Function

Logistic Regression

$[w_0 \ w_1 \ .. \ w_n]^T$

$\hat{y} = \sigma(w \cdot x) \rightarrow [1 \ x_1 \ .. \ x_n]$

$J = ?$

$x \longrightarrow \hat{y}$

y - Ground Truth } Cost

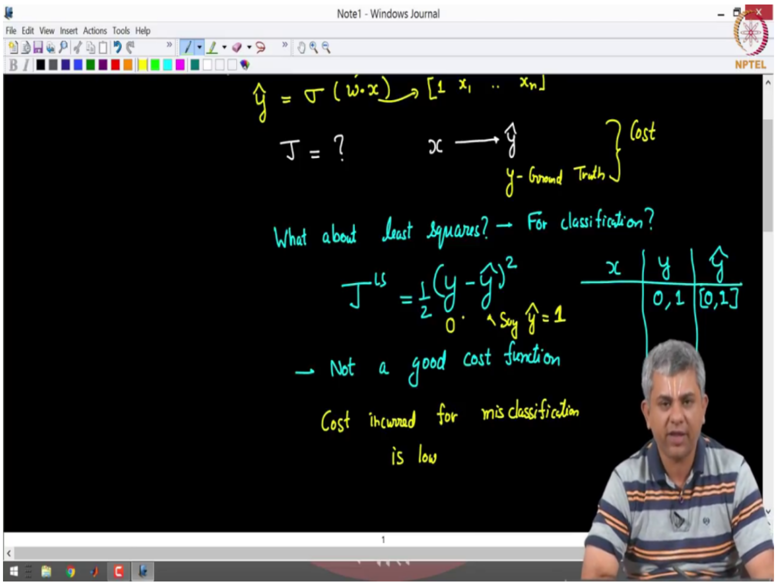
What about least squares? → For classifical

$$J^{LS} = \frac{1}{2}(y - \hat{y})^2$$

In the last video we saw that we had a simple algorithm or a forward model for logistic regression. We call that the forward model was Y hat = Sigma of W. X where W includes W not, W 1 up till W n if you have n features, and X here is one, X 1, up till X n so this was our forward model. Now the question is what is a good cost function for this? Remember, in our usual learning paradigm what we have is I have an X, this predicts Y hat, the ground truth is some Y and I wish to find out some cost or penalty for Y hat and Y being different ok.

So now why not use the least square cost function? The least square cost function was simply Y minus Y hat square divided by 2, of course I'm taking this for one particular incident or one particular example, the usual thing was we take the sum of all these examples and take an average that is what we did for linear regression, so why not use this for classification. It turns out that this is not a good model okay.

(Refer Slide Time: 2:18)



$\hat{y} = \sigma(\underbrace{w \cdot x}) \rightarrow [1 \; x_1 \; .. \; x_n]$

$J = ?$ $\quad x \longrightarrow \hat{y}$, y - Ground Truth $\}$ Cost

What about least squares? → For classification?

$J^{LS} = \frac{1}{2}(\underset{0}{y} - \hat{y})^2$ ↖ Say $\hat{y} = 1$

x	y	$\hat{y}$
	0,1	[0,1]

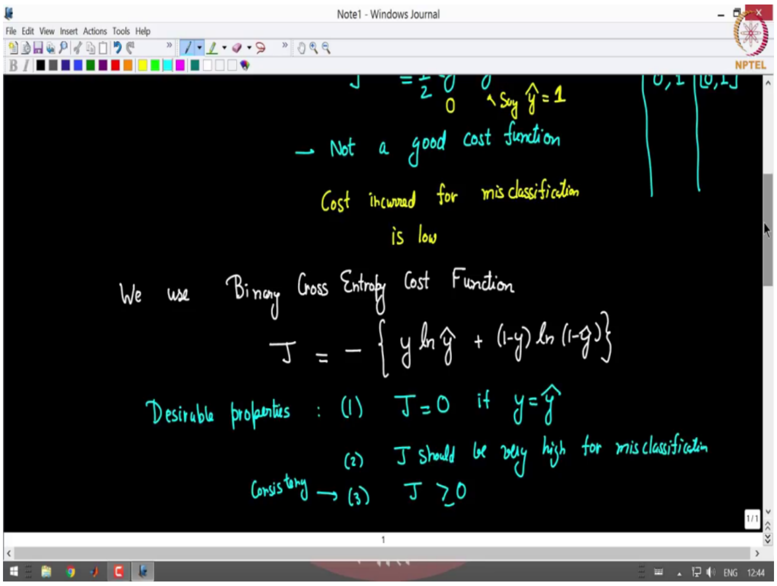
→ Not a good cost function

Cost incurred for misclassification is low

It is not a good cost function or at least it is not an optimal cost function for several reasons, I will just mention 1 okay. Now if you recall in a binary classification problem for a given X, underground truth Y and even Y hat all of these Y is either 0 or 1, Y hat lies between 0 and 1 okay it is some decimal number. Now suppose you are doing a case where you are trying to distinguish between let us say something as serious as cancer and no cancer or even if it is a cat versus a dog. Now, notice that even if you totally misclassify ok, so for example Y is 0 and let us say Y hat is 1 okay, so you are totally misclassifying for example, a case where the actual prediction or the ground truth is the person does not have cancer and you say it is cancer.

The cost that you incur for misclassification that is when Y is 0, if you say Y hat is 1 or close to 1 let us say 0.99, we saw in the previous video that Y hat gives an estimate of probability that the prediction is actually or the class is actually 1. So when we want to predict something as clear as classification and you give a misclassification, the cost incurred for that is actually very low okay that is we do not penalize this cost high enough okay, even though there is a penalty it is not high enough, so because of that that is one of the reasons why the usual least square cost function is a bad cost function for classification.

(Refer Slide Time: 4:22)



0 ↖ Say $\hat{y} = 1$

→ Not a good cost function

Cost incurred for misclassification is low

We use Binary Cross Entropy Cost Function

$$J = -\left\{ y \ln \hat{y} + (1-y) \ln (1-\hat{y}) \right\}$$

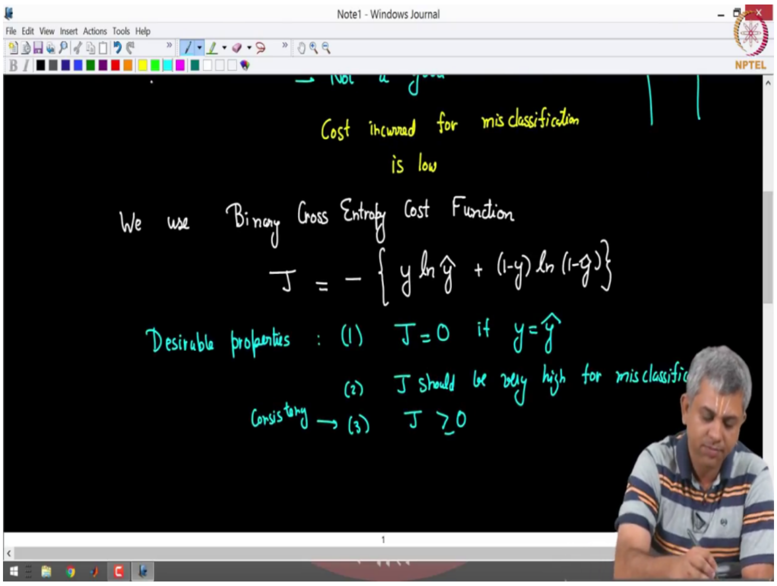
Desirable properties : (1) $J = 0$ if $y = \hat{y}$

(2) J should be very high for misclassification

Consistency → (3) $J \geq 0$

We instead use something called the Binary cross entropy cost function, the form of that cost function is different so J is there is a negative outside minus Y times l n Y hat + 1 minus Y times l n 1 minus Y hat, now we will come to the reasons for each of these terms shortly including why there is a minus and why both these terms are sitting there okay. So now let us think about some properties of the cost function that we want to have and let us check whether this has it or not. So some desirable properties for classification cost function, First of course is if J should be 0 if Y is equal to Y hat, this is the 1st thing that we have to check okay. Second, J should be very high for misclassification, and the 3rd this is merely required for consistency is that J should be greater than equal to 0.

(Refer Slide Time: 6:16)



→ Not a good

Cost incurred for misclassification is low

We use Binary Cross Entropy Cost Function

$$J = -\left\{ y \ln \hat{y} + (1-y) \ln (1-\hat{y}) \right\}$$

Desirable properties : (1) $J = 0$ if $y = \hat{y}$

(2) J should be very high for misclassific

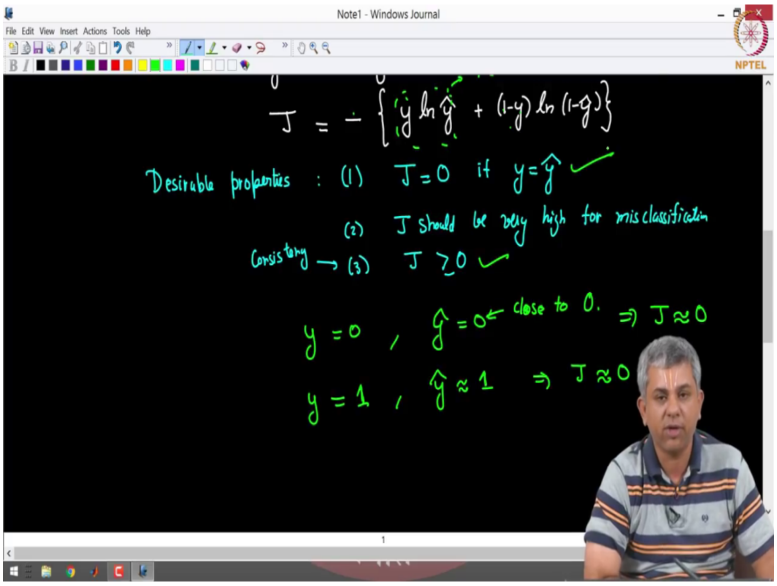
Consistency → (3) $J \geq 0$

Remember when we had our least square cost function; least square is obviously always positive ok so let us take this step-by-step. I will start with here, notice Y hat will always lie between 0 and 1, it is a probability okay so instead of saying that this person has cancer or not, you will say something like the probability that this person has cancer is 0.9 that is going to be the outcome of your logistic regression. Why is that? Because if you notice our Y hat is Sigma of something and the Sigma always goes between 0 and 1 because of that Y hat is constrained to be between 0 and 1 ok.

Notice that Y is either 0 or 1, it is not between 0 and 1 but it is either 0 or 1, why is that? Because why is the ground truth, ground truth we already know, ground truth this is a supervised learning task, you already know whether this person has cancer or not, this is X history or if you are trying to classify images let us say cat and dog, you have already available set and it is based on that label set that you are training so you already have all these labels available okay.

So Y is either 0 or 1, Y hat is between 0 and 1 therefore l n of Y hat is going to be negative okay, Y is going to be either 0 or 1 so this whole term is negative, similarly this term is also negative and that is why we have this minus sign so that the whole term actually becomes positive; negative multiplied by negative this is positive so this is the function of the minus sign, it is just to make J positive so that it is consistent with least squares okay that is the first. Now let us see, is J equal to 0 if Y equal to Y hat? Okay so let us take a few cases.

(Refer Slide Time: 8:04)



$$J = -\left\{ y \ln \hat{y} + (1-y)\ln(1-\hat{y}) \right\}$$

Desirable properties : (1) $J = 0$ if $y = \hat{y}$ ✓

(2) J should be very high for misclassification

Consistency → (3) $J \geq 0$ ✓

$y = 0$, $\hat{y} = 0$ ← close to 0. $\Rightarrow J \approx 0$

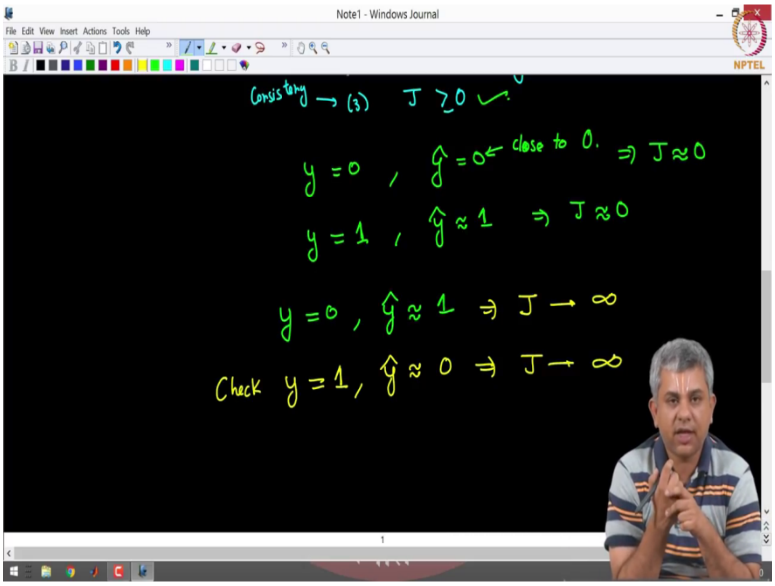
$y = 1$, $\hat{y} \approx 1$ $\Rightarrow J \approx 0$

Y is 0 and Y hat is 0 okay, now the moment Y is 0 and Y hat is 0 or close to 0, I will keep it close to 0 just to avoid the similarity at exactly 0, why am I keeping it close to 0 because Sigma is actually never going to give you exactly 0, remember Sigma is 1 + or Sigma of Z is 1 by 1 + E to the power minus Z okay. So in case Y is 0 and Y hat is close to 0 let us say 10 power minus 6 then this term is close to 0 and this term is again going to become close to 0, why is that? This is 1 and 1 minus Y hat this becomes l n of 1 so this becomes 0 ok, J is approximately 0.

Similarly you can check, if Y is 1 and Y hat is close to 1 okay then this term is exactly 0 and this term is going to be 1 multiplied by l n of approximately 1 so this is also going to be approximately 0 okay. So this condition is also satisfied that is if you classify correctly, your cost function is going to be approximately 0 ok. Third which is the main property which so these 2 properties are true of least squares also but this property is the one that least square does not satisfy okay, so what we want is J should be quote unquote high ok, in case you have misclassified so let us check that, I will check it just for one case.

So let us say, Y is 0 but Y hat is approximately 1 let us say 0.99 something or that sort. So just to give you an example, suppose the person does not have cancer or the image is let us say a dog and you end up saying that this is actually not a dog and I am very-very certain about it okay, I am certain up till 99.99% that this is actually a cat okay so you are actually misclassifying with high probability, then what happens to the cost function? So let us take a look, let us come here.

(Refer Slide Time: 10:27)



Consistency $\rightarrow$ (3) $J \geq 0$ ✓

$y = 0, \quad \hat{y} = 0 \leftarrow$ close to 0. $\Rightarrow J \approx 0$

$y = 1, \quad \hat{y} \approx 1 \Rightarrow J \approx 0$

$y = 0, \quad \hat{y} \approx 1 \Rightarrow J \rightarrow \infty$

Check $y = 1, \quad \hat{y} \approx 0 \Rightarrow J \rightarrow \infty$

So in this case, Y was 0 so this term actually becomes 0 because Y hat is close to 1 and this term becomes actually 0 because Y is actually 0. What happens to this term? So Y is 0, which means the coefficient here is approximately 1 and Y hat is approximately 1 which makes this term approximately 0. And what is l n 0? L n of 0 is minus infinity so we have got a minus sign here and you are going to throw up a really high cost because you have misclassified okay so that is the trick that we are using. J tends to infinity as Y hat tends to 1, also you can check as an exercise that if Y is 1 so the ground truth is 1 and if Y hat is approximately 0, J will again tend to infinity ok.

So the basic trick here is that in case you have a misclassification, you are going to throw up a very high cost and in case you have a correct classification, you are going to get J equal to 0 or approximately equal to 0 dependent on how close you are to correct classification okay, so this is what is called binary cross entropy cost function. So along with the least square function these two are the main two lost functions that we will be using more or less throughout the course okay, we will have a small modification to the binary cross entropy cost function shortly then we take multiclass classification but then it is a very minor adjustments to what actually is this okay.

So just two cost functions, therefore regression typically we use a regression type problem we typically use something like a least square cost function and for classification problem we primarily use something like a binary cross entropy cost function, thank you.