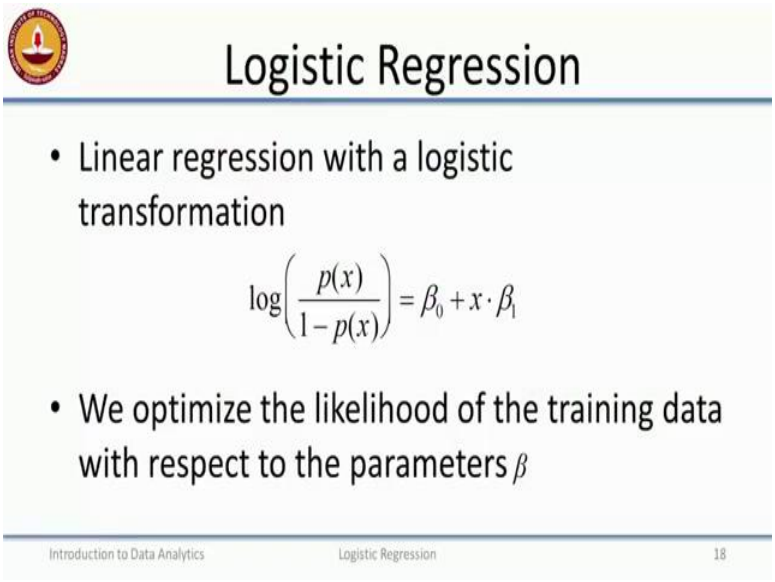
Introduction to Data Analytics
Prof. Nandan Sudarsanam
and Prof. B. Ravindran
Department of Management Studies
and Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 05
Lecture - 24
Training a Logistic Regression Classifier

Hi, so we are looking at the module on Training Logistic Regression Classifier now.

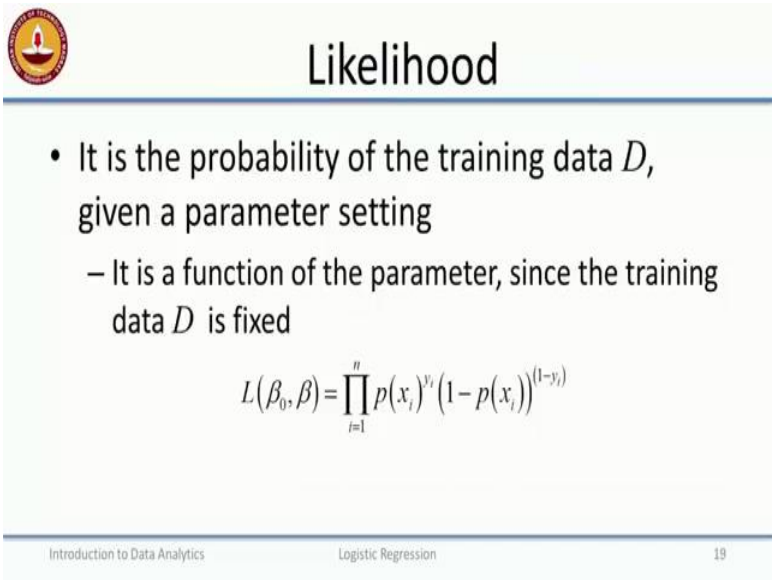
(Refer Slide Time: 00:19)



So, in the previous module we looked at the basic idea behind logistic regression, which is essentially to do linear regression with a logistic transformation. So, you took the log odds function, which is $\frac{p(x)}{1-p(x)}$, took the logarithm of it and try to fit a linear curve to this transform function. So, how do we find these parameters β, β_1 and β_0?

So, we optimize the likelihood of the training data with respect to the parameters β, so that is essentially the way we are going to be training this. So, this is slightly different from some of the earlier methods we have looked at identifying the parameters, mainly because we are looking at here the probability of classification, not just getting the classifications right or wrong, but we are actually looking at the probability of classification, that makes more sense to try to optimize the probability of seeing the training data with respect to the parameter β.

(Refer Slide Time: 01:17)



So, what is the likelihood? So, the likelihood is the probability of a training data D given a particular parameter setting β. So, you should note here that, it is the function of the parameter setting, because the training data D that is given to you is usually fixed. So, here is an example of the likelihood of some kind of classification tasks. So, I am going to assume that the data is given to you in the form of x_i, y_i pairs as we have done in the past.

So, for each x_i there is going to be an outcome y_i, which will be either 1 if it belongs to class 1 or 0 if it belongs to class 2. So, let us look at one term in the product that I have written down there, is you can see if the output corresponding to x_i is 1, then the first term in the product will be $p(x_i)$ and the second term in the product will be 1, because y_i is 1 and 1 - $p(x_i)$ is going to raise to the power of 0, which essentially reduce to 1.

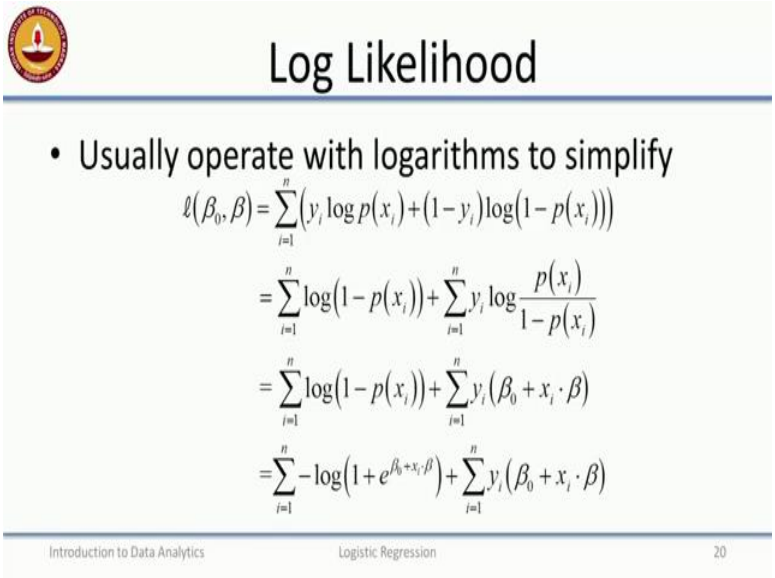
Likewise, if the output corresponding to x_i is 0, then the first term in the product is going to be 1 and the second term in the product will remain as 1 - $p(x_i)$. So, this essentially means that depending on, what the output variable is I am going to either take the probability of the data point occurring, probability of the data point having a label of 1 or the probability of the data point having the label of 0. So, to do recall that $p(x_i)$ is the probability that y = 1 for given x_i.

So, now, for this is for one data point and if I want to look at the probability of the entire data, I just take the product over all the data points, so the product runs from 1 to n. So, this expression now gives me the probability of seeing the training data given a specific

parameter setting. So, where do β and β_0 appear on the expression on the right hand side, so $p(x_i)$ is specified in terms of β and β_0.

So, implicitly, so β and β_0 are appearing on the right hand side of the equation and like I said, likelihood is the function of the parameters and hence we denote it as L of β. So, now our goal is to optimize this likelihood and, so that, so we get a good estimate of the parameter, so we have to find the right set of β, so that this probability is maximized. So, now, we look the term, the term looks a little hard to optimize, because lot of products here and, so we have to be little careful.

(Refer Slide Time: 04:16)



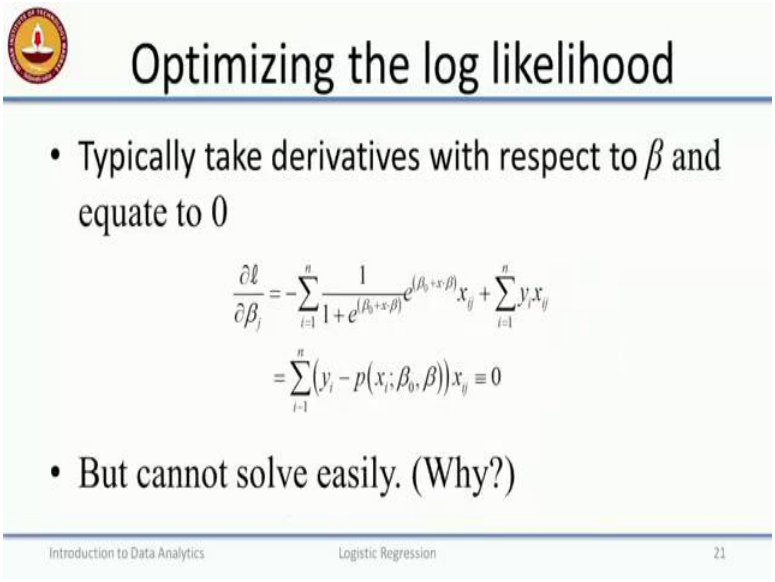
So, the usual way that we operate here is to take logarithms of this likelihood and you can see that lower case l here is used to denote the log of the likelihood function and then, you just walk through this log likelihood expression a little slowly. So, now, that I have taken the logarithms in the first step. In the first step, so I have taken logarithms and therefore, the products that I had earlier have become summations and the exponentiation that I had earlier have become products. So, that corresponds the original exponentiation I had in my expression and now they have become products.

Now, we can do a little bit of simplification here and you can see that, what I have essentially done is taken the… In the second step I have expanded the product term in the second term in this summation and then, I have gathered terms together which have a coefficient of y_i. So, that gives me the second term in the summation and the first term is just essentially 1*log(1 - $p(x_i)$). So, we know what log $\frac{p(xi)}{1-p(xi)}$ is and that is essentially

the function that you are trying to fit from the beginning.

So, we replace that with our linear fit that we had, the linear regressions fit that we did. And then, we do further simplification in order to come up with the expression given on the last line, that essentially writing out $p(x_i)$ and then, evaluating 1 - p (x_i) and that gives me the negative logarithms term on the last line of the expressions. So, now, what do we do? We have the log likelihood, so what we do to maximize this log likelihood.

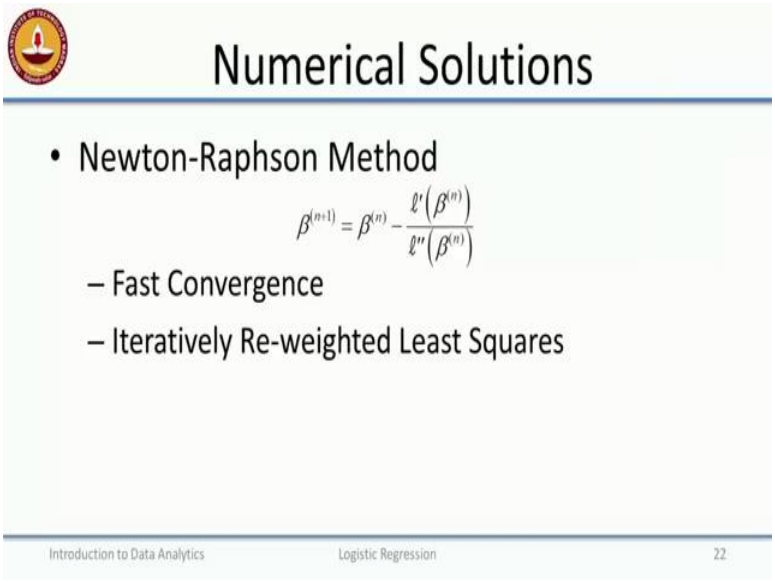
(Refer Slide Time: 06:01)



We essentially take the derivatives of this log likelihood with respect to β and then, we should be equating this to 0. So, the first line here is essentially taking the derivative of the log likelihood and it simplify to a very nice form, which is $y_i - p(x_i)^* x_i$ each individual component of x_i and now, we set this equal to 0. But, we really cannot solve this that easily. Why? Because, $p(x_i)$ is actually a transcendental function, so it is not very easy to find the close form solution for these kinds of expressions.

So, we have to actually look at numerical methods for solving these kinds of optimization problems and we essentially look at class of algorithms, which are known as interior point methods. So, I am not really going to get into the math behind all of this, but I assume that many of you have actually come across very simple optimization technique called Newton Raphson method.

(Refer Slide Time: 07:04)

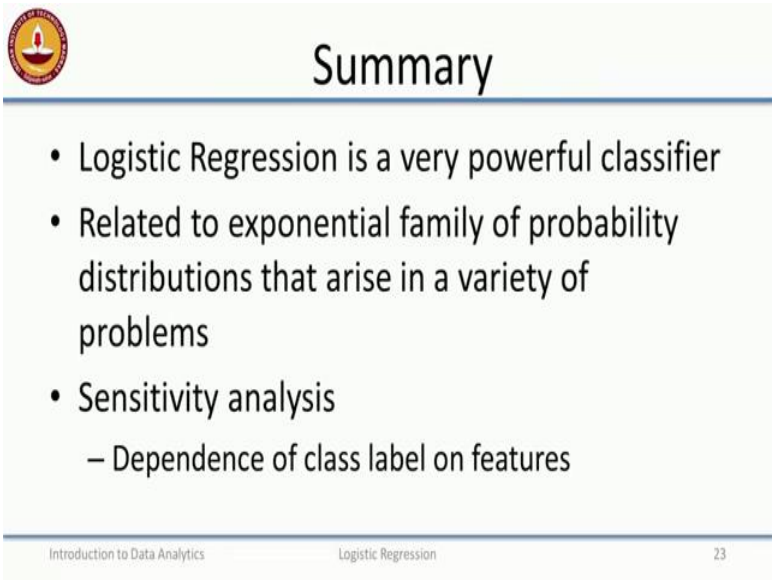


So and here is what the expression for Newton Raphson method is going to look like. So, I start off with a guess for my initial solution, the β and start of the guess β_0 and I would typically like my β_0 to be close to the true solution. And once I have the guess for β_0, then I keep updating the solution by essentially subtracting the first order derivative of the likelihood divided by the second order derivative of the likelihood. Take the ratio and then, subtract it from β in order to give me my next estimate.

So, this has fast convergence under certain regularity condition, so for one thing is that second derivative should like this and should be positive. And as you can see the first derivative is going to be 0 you are not going to be changing the value of your guess and when would the first derivative be 0 it will be 0 at one of the optima whether it is the maxima or the minima and you will also like this since the second derivative is going to positive to approach the minima. So, you can till when you approach optima you can be sure there is going to be the minima.

So, this is essentially the basic idea behind the Newton Raphson method and when Newton Raphson method is applied specifically to the logistic regression problem you come up with the iterative technique, which is called iterative re weighted least squares approach for training for finding the parameters in logistic regression. And, so most of these statistical packages that we have especially R in particular of front trust was have a very simple function that loves you to fit logistic regression to any data set that you have and they will essentially be using Newton Raphson by way of iterative re weighted least squares techniques.

(Refer Slide Time: 09:02)



To summarize this couple of modules logistic regression, so logistic regression is very powerful classifier build on the idea of doing linear regression on a logistic transformed output variables and the logistic regression is related to exponential family of probability distribution that rise in the variety of problems and that is the one of the reasons that make them very, very popular classifier.

And apart from that they work really well I mean, so that is the another reason that logistic regression is classifier of choice for many people's especially in medical domains, because they allow you to perform what you know as sensitive analysis. So, you can look at dependence of class labels on features by looking at the, the regression coefficient of specific feature in the fit that you obtain.

Thank you.