

Python for Data Science
Prof. Ragunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 35
Logistic regression

(Refer Slide Time: 00:23)

Data science for Engineers

Introduction

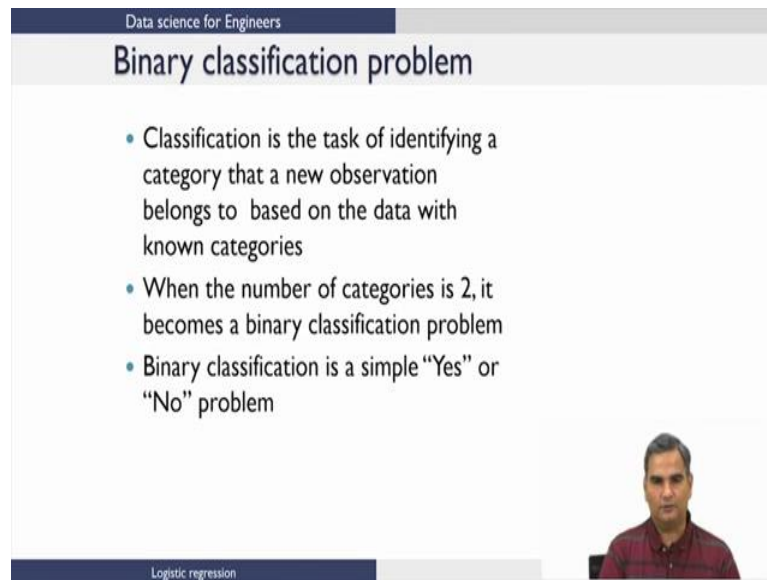
- Logistic regression is a classification technique
- Decision boundary (generally linear) derived based on probability interpretation
 - Results in a nonlinear optimization problem for parameter estimation
- Goal: Given a new data point, predict the class from which the data point is likely to have originated

Logistic regression 2

In this lecture I will describe a technique called Logistic regression. Logistic regression is a classification technique which basically develops linear boundary regions based on certain probability interpretations. While in general we develop linear decision boundaries this technique can also be extended to develop non-linear boundaries using what is called polynomial logistic regression. For problems where we are going to develop linear boundaries the solution still results in a non-linear optimization problem for parameter estimation as we will see in this lecture.

So, the goal of this technique is given a new data point I would like to predict the class from which this data point could have originated. So, in that sense this is a classification technique that is used in a wide variety of problems and it is surprisingly effective for a large class of problems.

(Refer Slide Time: 01:24)



Data science for Engineers

Binary classification problem

- Classification is the task of identifying a category that a new observation belongs to based on the data with known categories
- When the number of categories is 2, it becomes a binary classification problem
- Binary classification is a simple "Yes" or "No" problem

Logistic regression

Just to recap things that we have seen before we have talked about binary classification problem before just to make sure that we recall some of the things that we have talked about before. We said classification is the task of identifying what category a new data point or an observation belongs to. There could be many categories to which the data could belong, but when the number of categories is 2 it is what we call as the binary classification problem.

We can also think of binary classification problems as simple yes or no problems where you either say something belongs to a particular category or no it does not belong to that category.

(Refer Slide Time: 02:13)

Data science for Engineers


Input features

- Input features can be both qualitative and quantitative
- If the inputs are qualitative, then there has to be a systematic way of converting them to quantities
 - For example: A binary input like a "Yes" or "No" can be encoded as "1" and "0"
- Some data analytics approach can handle qualitative variables directly

$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$

Yes	No
0.1	0.05
0.3	-2

Logistic regression



Now, whenever we talk about classification problems we have described this before we say it data is represented by many attributes x_1 to x_n . We can also call this as input features as shown in the slide and these input features could be quantitative or qualitative. Now, quantitative features can be used as they are.

However, if we are going to use a quantitative technique, but we want to use input features which are qualitative, then we should have some way of converting this qualitative features into quantitative values. One simple example is if I have a binary input like a yes or no for a feature. So, what do we mean by this? So, I could have yes let us say a 0.1, 0.3 and then another data point could be no 0.05 - 2 and so on.

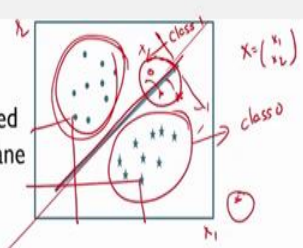
So, you notice that these are quantitative numbers while these are qualitative features. Now, you could convert this all into quantitative features by a coding yes as 1 and no as 0. So, then those also become number. There is a very crude way of doing this there might be much better ways of coding qualitative features into quantitative features and so on. You also have to remember that there are some data analytics approach that can directly handle these qualitative features without a need to convert them into numbers and so on. So, you should keep in mind that that is also possible.

(Refer Slide Time: 04:01)

Data science for Engineers

Linear classifier

- Decision function is linear
- Binary classification can be performed depending on the side of the half-plane that the data falls in
- We saw this before in the linear algebra module
- However, simply guessing "yes" or "no" is pretty crude
- Can we do something better using probabilities?



Logistic regression

Raghunathu

Now, that we have these features we go back to our pictorial understanding of these things just for the sake of illustration, let us take this example where we have this 2 dimensional data. So, here we would say x is x_1, x_2 variables let us say x_1 is here x_2 is here. So, its organized into data like this now let us assume that all the circular data belong to 1 category and all the stored data belong to another category. Notice that circle data would have certain x_1 and certain x_2 and similarly started I would have certain x_1 and certain x_2 .

So, in other words all values of x_1 and x_2 such that the data is here belongs to 1 class and such that the data is here belongs to another class. Now, if we were able to come up with hyper plane such as the one that is shown here we learn from our linear algebra module that to one side of this hyper plane is a half space this side is half space and depending on the way the normal is defined you would have a positive value and a negative value to each side of the hyper plane. So, this is something that we are dealt with in detail in 1 of the linear algebra classes.

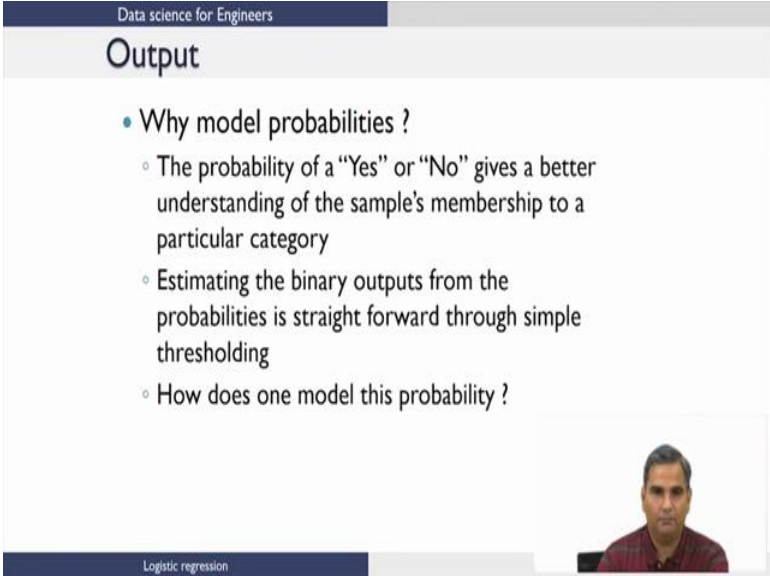
So, if you were to do this classification problem then what you could say is if I get a data point somewhere here I could say it belongs to whatever this class is here. So, let us call this for example, we could call this class 0 we could call this class 1 and we would say whenever a data point falls to this side of a line then it is class 0 and if a data point falls to this side of the line we will say its class 1 and so on.

However, notice that any data point, so whether it falls here it falls here we are going to say class 0, but intuitively you know that if this is really a true separation of these classes, then this is for sure going to belong to class 0. But as I go closer and closer to this line there is this uncertainty about whether it belongs to this class or this class because data is inherently noisy. So, I could have a data point which is true value here; however, because of noise it could slip to the other side and so on.

So, as I come closer and closer to this then you know the probability or the confidence with which I can say it belongs to a particular class can intuitively come down. So, simply saying yes this data point and this data point belongs to class 0 is 1 answer, but that is pretty crude. So, the question that this logistics regression and services can we do something better using probabilities.

So, I would like to say that the probability that this belongs to class 1 is much higher than this because its far away from the decision boundary. So, how do we do this? Is a question that logistics regression addresses.

(Refer Slide Time: 07:41)



Data science for Engineers

Output

- Why model probabilities ?
 - The probability of a "Yes" or "No" gives a better understanding of the sample's membership to a particular category
 - Estimating the binary outputs from the probabilities is straight forward through simple thresholding
 - How does one model this probability ?

Logistic regression

So, as I mentioned before the probability of something being from a class if we can answer that question that is better than just saying yes or no answers right. So, one could say yes this belongs to a class a better nuanced answer could be that yes it belongs to a class, but with a certain probability as the probability is higher, then you feel more confident about assigning that class to the data.

On the other hand if we model through probabilities we do not want to lose binary answer like yes or no also. So, if I have probabilities for something I can easily convert them to yes or no answers through some thresholding which we will see in the logistics regression methodology when we describe that. So, while we do not lose the ability to categorically say if a data belongs to a particular class or not by modeling this probability. On the other hand we get a benefit of getting a nuanced answer instead of just saying yes or no.

(Refer Slide Time: 08:50)

Data science for Engineers

Linear and log models

- Make $p(x)$ a linear function of x

$$p(x) = \beta_0 + \beta_1 X$$

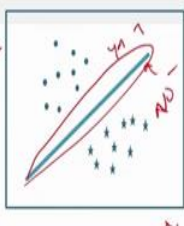
Handwritten note: $\beta_0 + \beta_{11}x_1 + \beta_{12}x_2$

 - This makes $p(x)$ unbounded below 0 and above 1
 - Might give nonsensical results making it difficult to interpret them as probabilities
- Make $\log(p(x))$ a linear function of x

$$\log(p(x)) = \beta_0 + \beta_1 X$$

Handwritten note: 0 and 1

 - Bounded only on one side



Logistic regression 7

So, the question then is how does one model these probabilities? So, let us go back and look at the picture that we had before let us say this is x_1 and x_2 . Remember that this hyper plane would typically have this form here the solution is written in the vector form, if I want to expand it in terms of x_1 and x_2 what I could do is I could write this as

$$p(x) = \beta_0 + \beta_{11}x_1 + \beta_{12}x_2.$$

So, this could be the equation of this line in this 2 dimensional space. Now, one idea might be just to say this itself is a probability and then let us see what happens. The difficulty with this is this p of x is not bounded because its just a linear function whereas, you know that the probability has to be bounded between 0 and 1. So, we have to find some function which is bounded between 0 and 1.

The reason why we are still talking about this linear function is because this is the decision boundary. So, what we are trying to do here is really instead of just looking at this decision

boundary and then saying yes and no + and -. What we are trying to do is we are trying to use this equation itself to come up with some probabilistic interpretation that is a reason why we are still sticking to this equation and trying to see if we can model probabilities as a function of this equation which is the hyper plane.

So, you could think of something slightly different and then say look instead of saying $p(x)$ is this let me say $\log(p) = \beta_0 + \beta_1 X_1$ in this case you will notice that it is bounded only on one side in other words if I write $\log(p) = \beta_0 + \beta_1 X_1$ will ensure that $p(x)$ never becomes negative.

However, on the positive side $p(x)$ can go to ∞ that again is a problem because we need to bound $p(x)$ between 0 and 1. So, this is an important thing to remember. So, it only bounds this one side. So, is that something more sophisticated we can do.

(Refer Slide Time: 11:13)

Sigmoid function

- Make $p(x)$ a sigmoid function of x

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

or $\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$

$f(x) = \frac{1}{1 + e^{-\beta x}}$

Sigmoid

- $p(x)$ bounded above by 1 and below by 0
- Good modeling choice for real life scenarios
- The LHS can be interpreted as the log of odds-ratio in the second equation

Logistic regression

The next idea is to write $p(x)$ as what is called a sigmoidal function; the sigmoidal function as relevance in many areas. So, this is a function that is used in neural networks and other very interesting applications. So, the sigmoid has an interesting form which is shown here.

Now, let us look at this form right here. I want you to notice two things; number 1 is still we are trying to stick this hyper plane equation into the probability expression because that is a decision surface. Remember intuitively somehow we are trying to convert that hyper

plane into a probability interpretation. So, that is the reason why we are still sticking to this $\beta_0 + \beta_1 X$.

Now, let us look at this equation and then see what happens. So, if you take this argument $\beta_0 + \beta_1 X$. So, that argument depending on the value of x you take could go all the way from $-\infty$ to ∞ right. So, just take a single variable case if I write let us say $\beta_0 + \beta_1$ just 1 variable X not a vector.

Now, if β_1 is positive if you take x to be a very very large value this number will become very large if β_1 is negative if you take x to very very large value in the positive side this number will become $-\infty$. And similarly if β_1 takes the other values you can correspondingly choose x to be positive or negative and then make this unbounded between $-\infty$ to ∞ .

So, we will see what happens to this function when $\beta_0 + \beta_1 X$ is $-\infty$ you would get this to be $e^{-\infty}$ divided by $1 + e^{-\infty}$ or you can just think of this as very large number. So, in that case the numerator will become 0 and the denominator will become $1 + 0$. So, on the lower side this expression will be bounded by 0.

Now, if you take $\beta_0 + \beta_1 X$ to be a very large positive number, then the numerator will be a very very large positive number and the denominator will be $1 +$ that very large positive number. So, this will be bounded by 1 on the upper side. So, now from the equation for the hyper plane we have been able to come up with a definition of a probability which makes sense which is bounded between 0 and 1. So, its an important idea to remember.

By doing this what we are doing is the following, if we were not using this probability all that we will do is we look at this equation and whenever a new point comes in we will evaluate this $\beta_0 + \beta_1 X$. And then based on whether its positive or negative we are going to say yes or no right.

Now, what has happened is instead of that this number is put back into this expression and depending on what value you get you get a probabilistic interpretation that is a beauty of this idea here. You can rearrange this in this form and then say

$$\log \left(\frac{p(x)}{1 - p(x)} \right) = \beta_0 + \beta_1 X$$

the reason why I show you this form is because the left hand side could be interpreted as log of odds ratio which is an idea that is used in several places. So, that is the connection here.

(Refer Slide Time: 14:56)

Data science for Engineers

Estimation of the parameters

- We find parameters in such a way that plugging these in the model equation should give the best possible classification for the inputs from both the classes
- This can be formalized by maximizing the following likelihood function

$$L(\beta_0, \beta_1) = \prod_{i=1}^n (p(x_i))^{y_i} (1 - p(x_i))^{(1-y_i)}$$

when x_i belongs to class 0, $y_i = 0$
 when x_i belongs to class 1, $y_i = 1$

Logistic regression 9

Now, we have this probabilities and remember if you were to write this hyper plane equation as the way we wrote in the last few slides $\beta_0 + \beta_{11}x_1 + \beta_{12}x_2$. The job of identifying a classifier as far as we are concerned is done when we identify values for the parameters β_0 , β_{11} and β_{12} . So, we still have to figure out what are the values for this.

Once we have a value for this any time I get a new point I simply put it into the p of x equation that we saw in the last slide and then get a probability right. So, this still needs to be identified. And; obviously, if we are looking at a classification problem where I have this on this side and starts on this side, I want to identify these β_0 , β_1 , β_{11} and β_{12} such a way that this classification problem is solved.

So, I need to have some objective for identifying these values right. Remember in the optimization lectures I told you that all machine learning techniques can be interpreted in some sense as an optimization problem. So, here again we come back to the same thing and then we say look we want to identify this hyper plane, but I need to have some objective function that I can use to identify these values.

So, these β_0 , β_{11} and β_{12} will become the decision variables, but I still need an objective function. And as we discussed before when we are talking about the optimization techniques the objective function has to reflect what we want to do with this problem. So, here is an objective function looks little complicated, but I will explain this as we go along.

So, I said in the optimization lectures we could look at maximizing or minimizing. In this case what we are going to say is I want to find values for β_0 , β_{11} and β_{12} such that this objective is maximized. So, take a minute to look at this objective and then see why someone might want to do something like this. So, when I look at this objective function let us say I again draw this and then let us say I have these points on one side and the other points on the other side.

So, let us call this class 0 and let us call this class 1. So, what I would like to do is I want to convert this decision function into probabilities. So, the way I am going to think about this is when I am on this line I should have the probability being equal to 0.5 which basically says that if I am on the line I cannot make a choice between class 1 and class 2 because the probability is exactly 0.5. So, I cannot say anything about it.

Now, what I would like to do is you can interpret it in many ways; one thing would be to say as I go away from this line in this direction I want the probability of the data belonging to class 1 to keep decreasing. The minute that the probability that the data belongs to class 1 keeps decreasing that automatically means since there are only 2 classes and this is a binary classification problem, the probability that the data belongs to class 0 keeps increasing.

So, if you think of this interpretation where as I go from here. So, here the probability that the data point belongs to class 1 let us say is 0.5, then basically it could either belong to class 1 or class 0. And if it is such that the probability keeps decreasing here of the data point belonging to class 1 then it has to belong to class 0. So, that is a basic idea.

So, in other words we could say the probability function that we define before should be such that whenever a data point belongs to class 0 and I put that into that probability expression I want a small probability. So, you might interpret that probability as the probability that the data belongs to class 1 for example. And whenever I take a data point from this side and put it into that probability function then I want the probability to be very

high because I want that is the probability that the data belongs to class 1. So, that is a basic idea.

So, in other words we can paraphrase this and then say for any data point on this side belonging to class 0, we want to minimize $p(x)$ when x is substituted into that probability function and for any point on this side when we substitute these data points into the probability function we want to maximize that probability. So, if you look at this here what this says is if this data point belongs to class 0 then y_i is 0.

So, whenever a data point belongs to class 0 anything to the power 0 is 1; so this will vanish. So, in the product there will be functions of this form which will be $1 - p(x_i)$ and because y_i is 0 this will become 1. So, this will become something to the power 0 1. So, this term will vanish and the only thing that will remain is $1 - p(x_i)$. So, if we try to maximize $1 - p(x_i)$, then that is equivalent to minimizing $p(x_i)$.

So, for all the points that belong to class 0 we are minimizing $p(x_i)$. Now, let us look at the other case of a data point belonging to class 1 in which case y_i is 1. So, $1 - 1^0$; so this term will be something to the power 0 which will become 1. So, it kind of drop out. So, the only thing that will remain is $p(x_i)$ now y_i is 1, so power 1 it will be just left with p of x_i . And since this data belongs to class 1 I want this probability to be very large. So, when I maximize this it will be a large number.

So, you have to think carefully about this equation there are many things going on here; number 1 that this is a multiplication of the probabilities for each of the data points. So, this includes data points from class 0 and class 1. The other thing that you should remember is let us say I have a product of several numbers, if I am guaranteed that every number is positive right then the product will be maximized when each of these individual numbers are maximized. So, that is a principal that is also operating here that is why we do this product of all the probabilities.

However, if a data point belongs to class 1 I want probability to be high. So, the individual term is just written as $p(x_i)$, so this is high for class 1. When a data point belongs to class 0 I still want this number to be high; that means, this number will be small. So, it automatically takes care of this as far as class 0 and class 1 are concerned.

So, while this looks little complicated this is written in this way because its easier to write this as 1 expression. Now let us take a simple example to see how this will look. Let us say I have class 0. I have 2 data points x_1 and x_2 and class 1. I have 2 data points x_3 and x_4 . So, this objective function when its written out would look something like this.

So, when we take let us say these points belonging to class 0, then I said the only thing that will be remaining is here. So, this will be $1 - p(x_1)$, for the second data point it will be $1 - p(x_2)$, then for the data third data point will be $p(x_3)$ and for the fourth data point it will be $p(x_4)$. So, this would be the expression from here.

So, now, when we maximize this then since p of x s are bounded between 0 and 1 this is a positive number; this is a positive number; positive number positive number and if the product has to be maximized, then each number has to be individually maximized; that means, this has to be maximized. So, it will go closer and closer to 1 the closer to 1 it is better.

So, you notice that to x_4 would be optimized to belong in class 1 similarly x_3 would be optimized to belong in class 1 and when you come to these 2 numbers you would see that this would be a large number if $p(x_1)$ is a small number. So, $p(x_1)$ basically means that x_1 is optimized to be in class 0 and similarly x_2 is optimized to be in class 0. So, this is an important idea that you have to understand in terms of how this objective function is generated.

(Refer Slide Time: 24:44)

Data science for Engineers

Log-likelihood function

- The log-likelihood function will become

$$l(\beta_0, \beta_1) = \sum_{i=1}^n y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i))$$
- Simplifying this expression and using the definition for $p(x)$ will result in an expression with the parameters of the linear decision boundary
- Now the parameters can be estimated by maximizing the above expression using any nonlinear optimization solver

Logistic regression
10

Now, one simple trick you can do is take that objective function and take a log of that and then maximize it. So, if I am maximizing a positive number x then that is equivalent to maximizing $\log(x)$ also. So, whenever this is maximized that will also be maximized. The reason why you do this it makes the product into some makes it looks simple. So, remember from our optimization lectures we said we are going to maximize this objective. So, we always write this objective in terms of decision variables and the decision variables in this case are β_0 , β_{11} and β_{12} as we described before.

So, what happens is each of these probability expressions if you recall from your previous slides will have these 3 variables and x_i are the points that are already given, so you simply substitute them into this expression. So, this whole expression would become a function of β_0 , β_{11} and β_{12} right.

Now, we have come back to our familiar optimization territory where we have this function which is a function of these decision variables this needs to be maximized and this is an unconstrained maximization problem because we are not putting any constraints on β_0 , β_1 and β_2 . So, they can take any value that we want. And also the fact that the way the probability is defined this would also become a non-linear function. So, basically we have a non-linear optimization problem in several decision variables and you could use any non-linear optimization technique to solve this problem.

And when you solve this problem what you get is basically the hyper plane. So, in this case its a 2 dimensional problem. So, we have 3 parameters now if there is an n dimensional problem if you have let us say n variables. So, I will have something like $\beta_0 + \beta_{11}x_1 + \beta_{12}x_2 + \dots + \beta_{1n}x_n$ this will be an $n + 1$ variable problem there are $n + 1$ decision variables. These $n + 1$ decision variables will be identified through this optimization solution. And for any new data point once we put that data point into the $p(x)$ function that sigmoidal function that we have described, then you would get the probability that it belongs to class 0 or class 1.

So, this is the basic idea of logistic regression. In the next lecture I will take a very simple example with several data points to show you how this works in practice and I will also introduce a notion of regularization which would help in avoiding over fitting when we do logistics regression. I will explain what over fitting means in the next lecture also. With that you will have a theoretical understanding of how logistics regression works and in a

subsequent lecture Doctor Hemant Kumar would illustrate how to use this technique in our on a case study problem.

So, that will give you the practical experience of how to use logistics regression and how to make sense out of the results that you get from using logistics regression on an example problem.

Thank you and I will see you in the next lecture.