Introduction to Proteogenomics
Dr. Sanjeeva Srivastava
Dr. David Fenyo
Department of Bioscience and Bioengineering
Indian Institute of Technology, Bombay

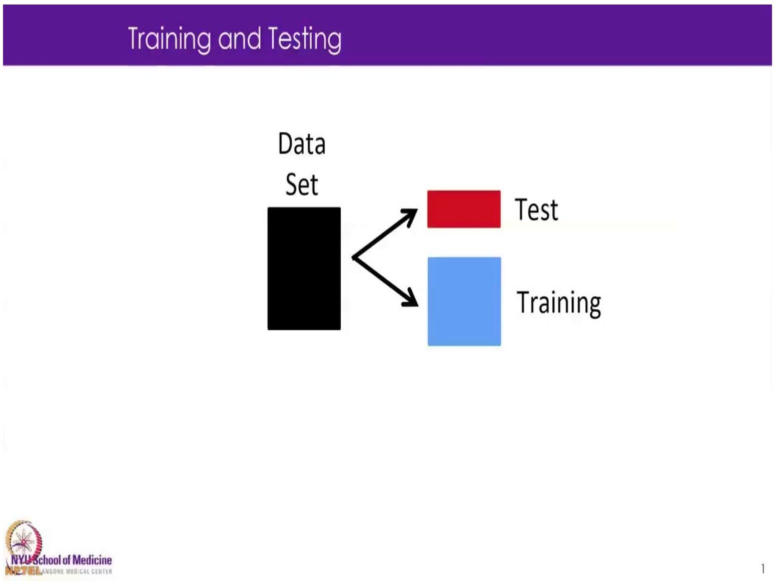
Lecture – 47
Predictive Analysis Part II

Welcome to MOOC course on Introduction to Proteogenomics. In today's lecture, Dr. David Fenyo will talk about training a model and test model. It will be continuation from the previous lecture where Dr. David Fenyo will briefly discuss testing error and training set size where he will also discuss about low variance and high variance.

He will provide a detailed idea about regularization and how regularization helps in training a model. He will then talk about how to divide dataset between training and test and how to deal with situation where your dataset is a small in number. I hope some of these discussions and points will be very important for you to consider when you are planning a big clinical study for your own project. So, let us welcome Dr. David Fenyo for today's lecture.

So, in another way to say this is that if you have a dataset you should divide it into tests and training set. Now, the problem for us again which is always the case that we do not have very much data. So, if we separate out a large chunk into the test set we do not have much left for training.

(Refer Slide Time: 01:53)



Training and Testing

Data Set

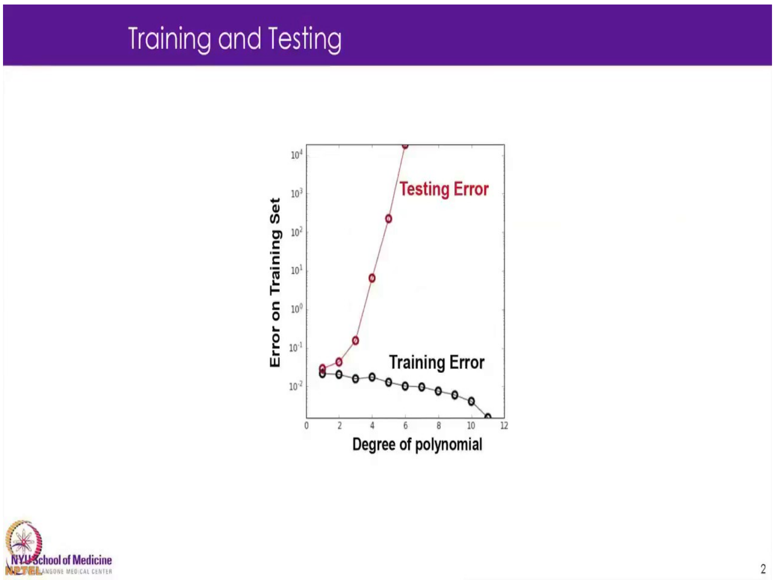
Test

Training

1

And, that is not good because I mean as we said before the larger your training dataset the better you will do, the better the model. So, now we are going to come back to this, but what people do is cross validation where they do this separation one way and then do with the separation another way and so on, but we will get back to that a little bit later.

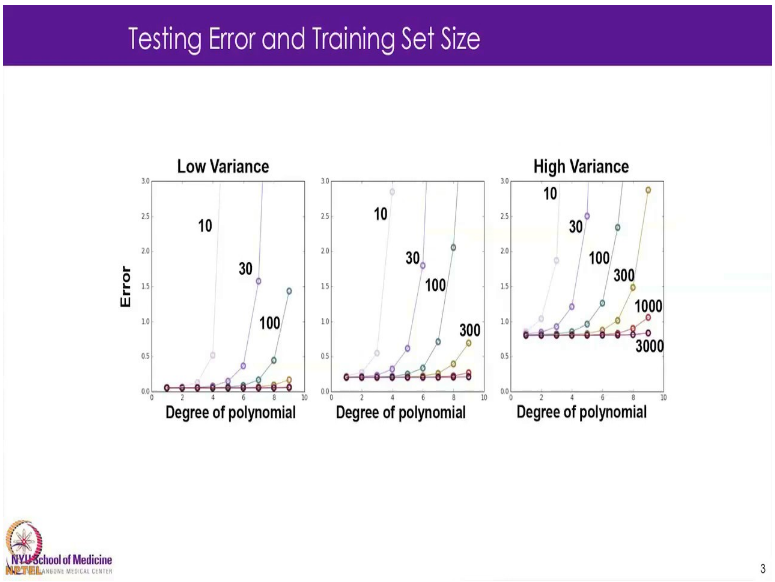
(Refer Slide Time: 02:19)



Training and Testing

So, now if we separate outs the training set. So, we saw now this is the same data, but the y axis is on a log scale. So, we saw that the training error can go down we can by making the model complex we can make it go to 0, but if we then compare it to our test sets that is the test error goes up as we make it make our model more complex.

(Refer Slide Time: 02:53)



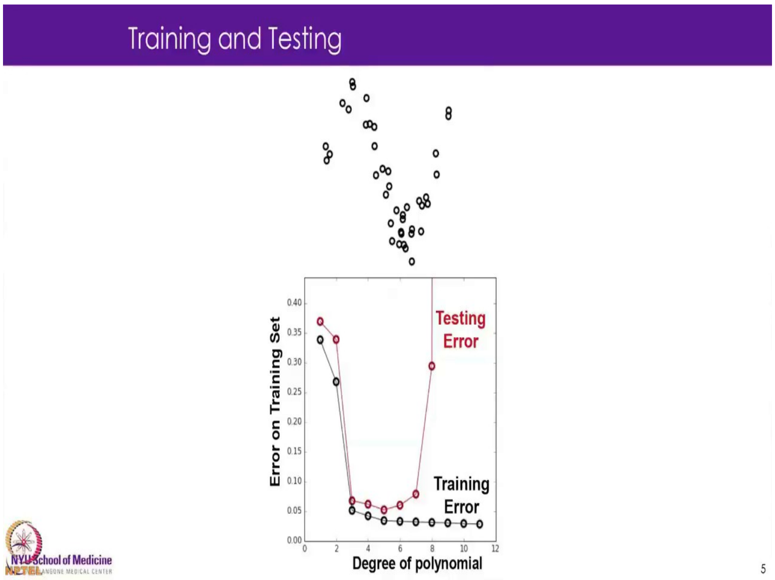
Testing Error and Training Set Size

And, so, one way the to get around that is if we really have a lot of data that helps. So, in this case now we are looking at the test error as a function of training data sets and as you see sorry the y-axis is not labeled, but as we increase so, the numbers there that are shown are the number of points that is we have.

So, you see that when we have few points like 10 we have we just increase our complexity the model complexity little bit the test error, but if we have large number of sample so, that the other two curves that are not labeled which have low errors are 1000 and 3000 points. So, that is one way of; so, the best way actually of doing what we will talk about later which is regularization to get around this problem is to have just very large data set.

So, I am going to just skip this ok. So, now, let us say we do not have now so far we have had lengthier dataset now we are going to have a little bit more complex dataset and we are going to do the same analysis.

(Refer Slide Time: 04:21)



Training and Testing

So, here now it is good to increase the complexity little bit because our data I mean as you can see we cannot fit that function well with the line. There is no way that we can get a good approximation and that is why we see that if we have a first degree polynomial which is a lot we do not and both 1 and 2 gives us very high errors, but then it goes down.

But, as also before in the example when we continue see when we continue increasing their complexity the training error goes down and down and the eventually evolved it is very low, but then at some point the testing error goes up. So, that is why we should choose the complexity somewhere here close to the minimum and it is usually better to be more conservative. So, I mean probably better not to choose 5 or 6, but rather 3 or 4 the. In this case that is sort of a rule of thumb that since we have this flat region it is better to be more conservative yeah.

Student: Sir, I ask you few questions. So, slit how to interview when you training error is going down with increasing polynomial or number of errors why should the testing error go up because your I mean training it well but, when you are testing it that error goes?

Yeah. So, I mean the reason is that you over train it; so, you over fit your model. So, you train it to the particular noise that is in your training set.

Student: Yeah.

So, it learns something that is not relevant to the process, but just because you have a finite set that you are training on that will have some by chance some noise and that is what you learn that noise and, but that does not generalize.

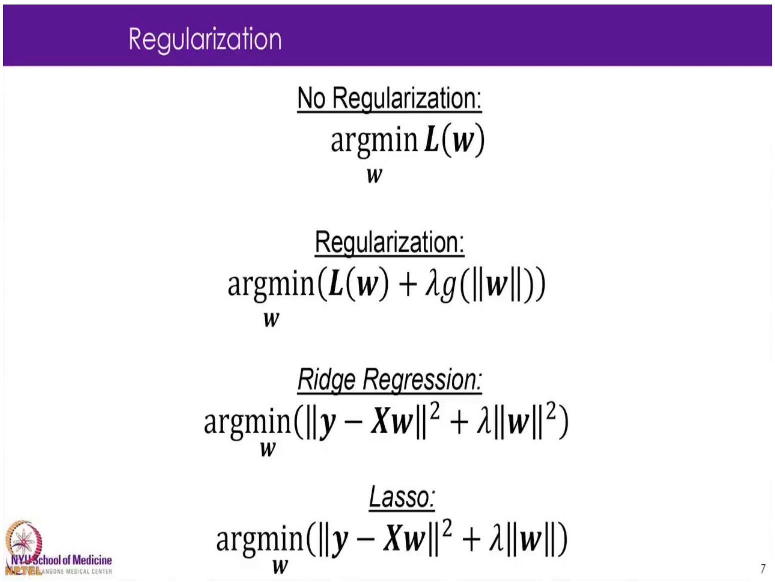
Student: How to determine it when the training is complete like we are not going to be over fit the over fit the training data+ the testing area it going to again like at what limit how we will we can dept like the training.

Yeah. So.

Student: We ask we are going to said actually this is not going to be over fitting again.

Yeah. So, we will talk a little bit about that I mean over fitting is always more even if you take these into account, but the way we are going to talk about that and usually we do it through cross validation that is, but we will get back to that later and this I am going to skip this.

(Refer Slide Time: 07:29)



Regularization

No Regularization:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}}\, \boldsymbol{L}(\boldsymbol{w})$$

Regularization:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}}(\boldsymbol{L}(\boldsymbol{w}) + \lambda g(\|\boldsymbol{w}\|))$$

Ridge Regression:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}}(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2)$$

Lasso:

$$\underset{\boldsymbol{w}}{\operatorname{argmin}}(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|)$$

7

So, now regularization; so, again Mani mentioned this. So, what we little bit formal way to that we are going to look at so.

I do not know how familiar you guys are with mathematical notations, but this is sort of what we do one way of describing what we do when we train a model. So, we have w or parameters. So, w here is bold. So, meaning that it is a vector. So, we have many values of the parameter. So, w 1, w 2 and so on and then we have a and a function L which we call the loss function and in the case of linear regression we mentioned that it is the sum of the square deviations that are a good loss function.

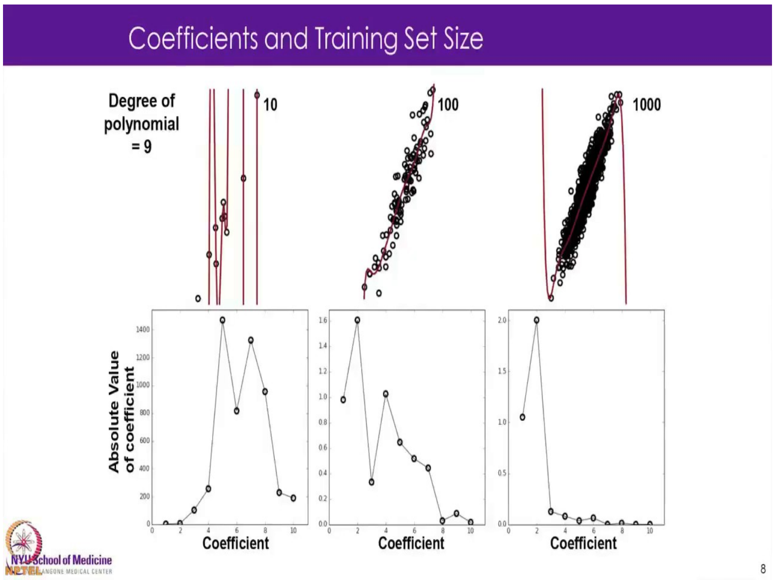
But, it is not necessarily the only one, there are other ones also you could also you do not need to take the square you could have the absolute add their up the absolute values. So, we choose some kind of loss function actually the L should not be bold sorry about that because it is a function with one it is has many a vector as an input, but output is just and then we try to find the w which minimizes the loss function.

So, that is why we call it least squares fit. So, the so, that is what we are doing, but again. So, we saw that is if we do this we and have many large vectors or many w's we run the risk of overfitting. So, what we can do is add an extra sum here where this is some kind of function of the absolute value of w. So, meaning that we are going to force w to be pretty small and then this parameter lambda is the one that governs that.

And, and there are two that are pretty off the news is either you add in the square of the length of w or you can and that is called ridge regression or you add in just absolutes with the length of w. So, those are two ways to regularize and minimize the risk of overfitting. But, again remember that even if we do all these things that we are going to talk about more thing that to try not to over fit; even if you do this there is a risk that we over fit and we should always be worried about the overfitting.

So, let us have a look at what happens here.

(Refer Slide Time: 10:45)



Coefficients and Training Set Size

So, now these are just 10 points here again we do a linear regression, but with a polynomial of degree 9 and it is a will be widely oscillating curve that is cut off over here. And, we see that coefficients are these are for a linear case it would be these two that is we would give us a line, but then we see that the other coefficients are very high.

Now, as I mentioned the best way to regularize is to have lots of data and if we have instead of 10 points 10 measurements, 100 measurements you see that we get reasonably good fit a little bit bigger and we see that it is. So, our most mainly dominated by the two first ones which would be aligned and if we get even more data it looks even better. But, outside the region of course, anything can happen where we do not have any data.

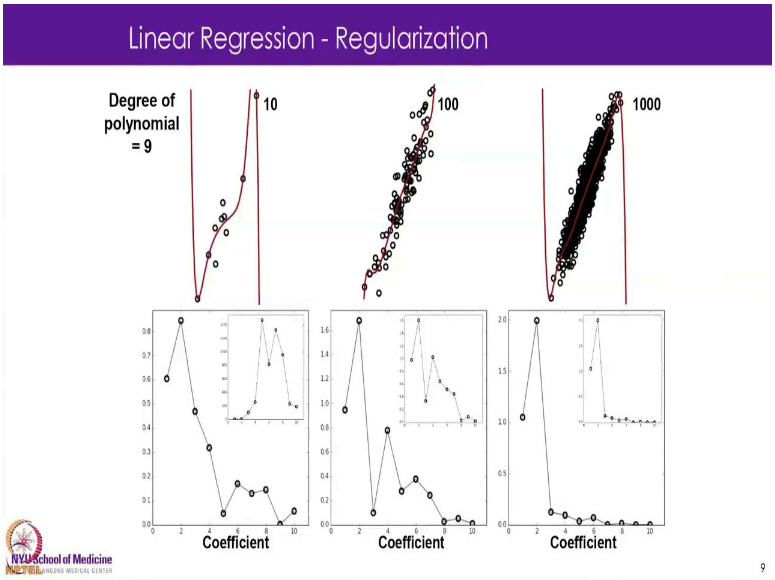
So, now if we look at the same thing we add some regularization and so, we instead of just minimizing the loss function we minimize the loss function plus a lambda times the in this case such believe it was the square of the parameters. So, then the same thing today here is the inserts. So, this is what we looked at previously without regularization those are the inserts and we see that with regularization we see that we get the much better fit than may it is dominated by the two first parameters in all the cases and So, that is we definitely need to do that.

Student: Sir, inside one is the regularized one you are saying?

Now, that the large sorry the last bases the regularized one and this is the same as I should un-regularized that is the same as the previous slide. So, if you look at this and we go I go back to the previous.

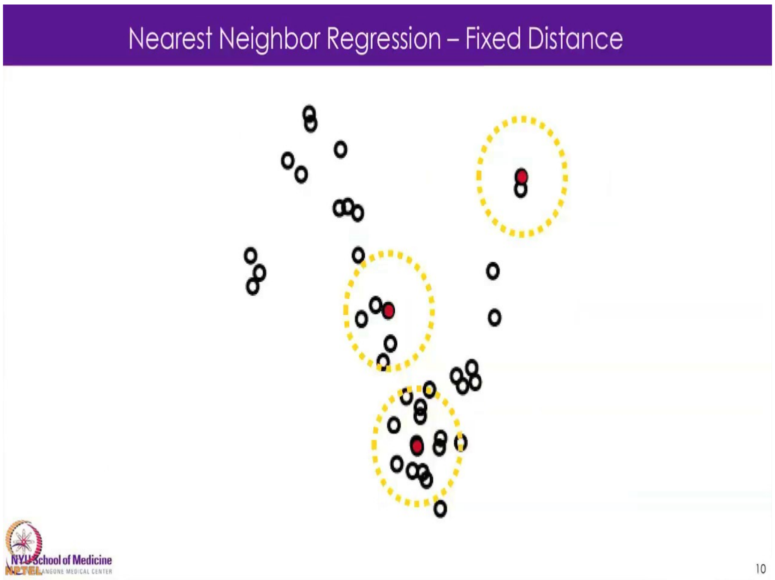
So, this one is the same as the insert in the next one ok.

(Refer Slide Time: 13:17)



Linear Regression - Regularization

(Refer Slide Time: 13:21)

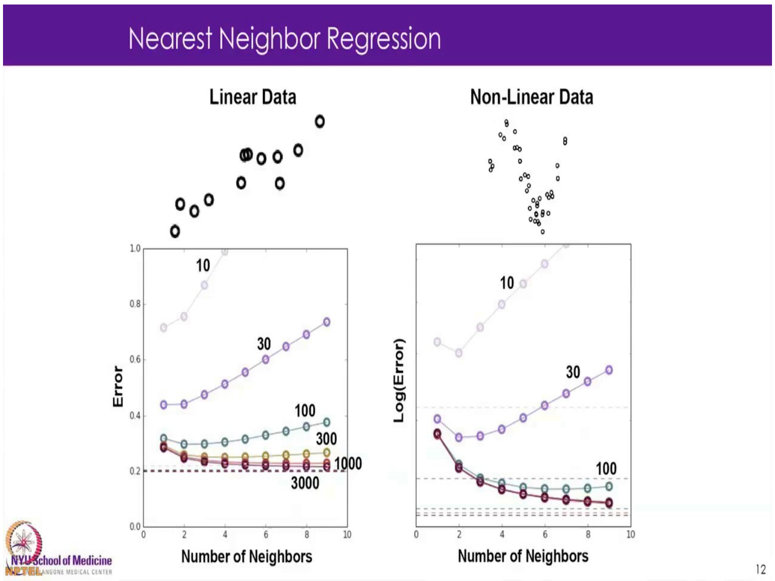


Nearest Neighbor Regression – Fixed Distance

So, then another way of doing it which is to do nearest neighbor regression; so, in this case we want to see what these red points out of ones we are interested in. So, for example, if we take the three nearest neighbors we would take an average of these values and approximate where the red ones would be. So, that is a the linear regression we have a very a sort of fixed model, but here it is really we just looking for data points that are similar.

So, it becomes can become very flexible, but often with high dimensions there are no points that are similar because it is. I do not know you should try to think about how a very high dimensional space looks like and it is it is not easy to think about. So, think about try to think about instead of in this case we have two dimensions just hundred dimensions. Something so, in 2-dimensions you often have points that are reasonably close not so much in this case, but in hundred even you spread out all your points in a much bigger space and it becomes like nothing is near anything else if you do not have an enormous amount of data which we usually do not have.

(Refer Slide Time: 15:09)

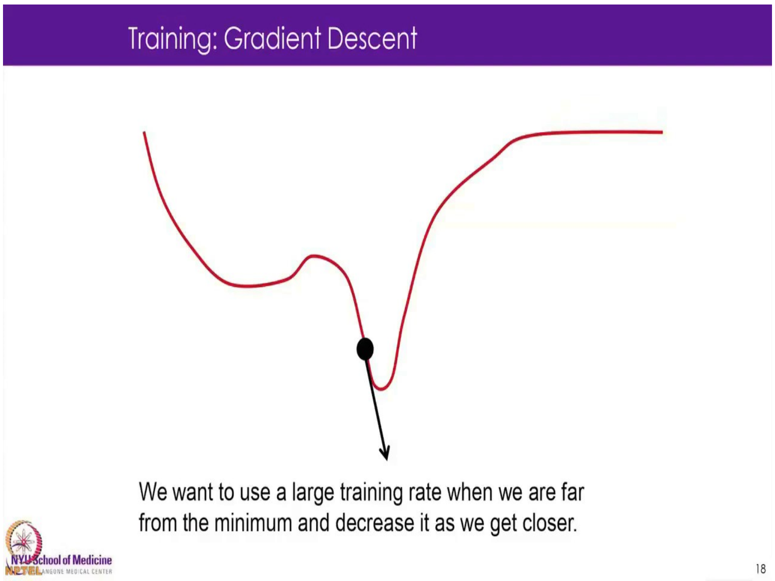


Nearest Neighbor Regression

Its often even if the linear model is not the we know that let us say the linear model is not right and it is often still better to assume a linear model because in most of the our cases we do not have that much data. So, that is another thing ok.

So, we looked at a little bit about model complexity now. Now, I will switch to how do we train the model. So, we already mentioned that we define a loss function which gives us a energy landscape that we try to find minima.

(Refer Slide Time: 15:49)



Training: Gradient Descent

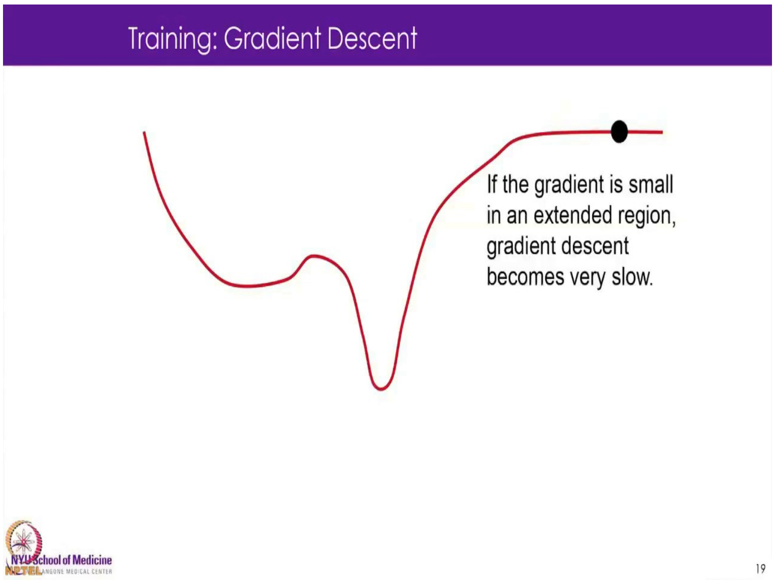
We want to use a large training rate when we are far from the minimum and decrease it as we get closer.

So, and usually we have our function defined and then a most often we start at a random place let us say here we just at first we just randomly assign our parameters and then what we want to do is to go from our randomly assigned space to the minimum. And, but we do not I mean and also we have this 10000 dimensional space that we have to work around them and. So, we, but what we know is the local environment.

So, in we what we can calculate is if we are here we know we can see what the slope is in which direction should we go through at least get further down and so, we calculate the derivative locally and then we go take a small step in the direction of down. And, so, then maybe we are go down then repeat this, take another step, go even further down and then continue again, but now what can happen when we get close to the minima is that we take too big over step.

So, we over jump and we are going to see that that is it is often good to start taking big steps and then at the end take smaller and smaller steps.

(Refer Slide Time: 17:23)

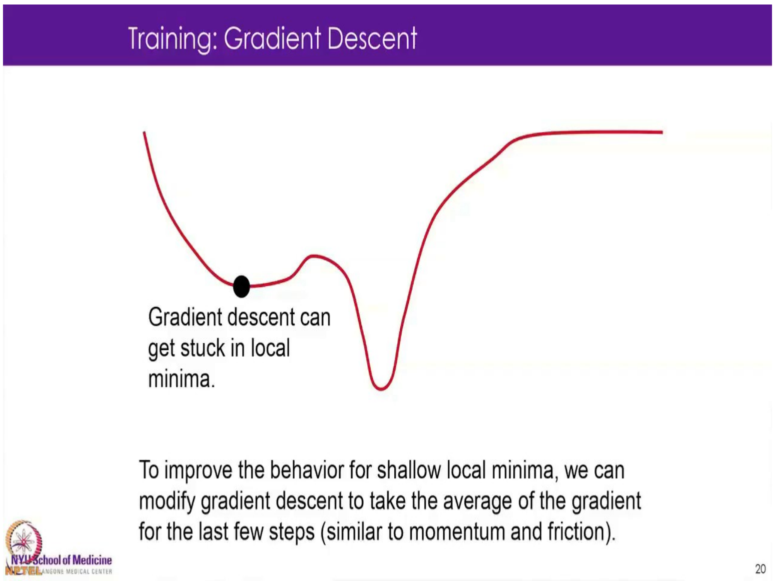


Training: Gradient Descent

If the gradient is small in an extended region, gradient descent becomes very slow.

Another problem that we can run into is that if we start in a region where it is very flat, there is no gradient almost or maybe not at all and then we and the stuck there since we always want to take a step in the direction of the gradient and the size of the step is also proportional to the gradient. So, then we have that is not that is a problem.

(Refer Slide Time: 17:53)



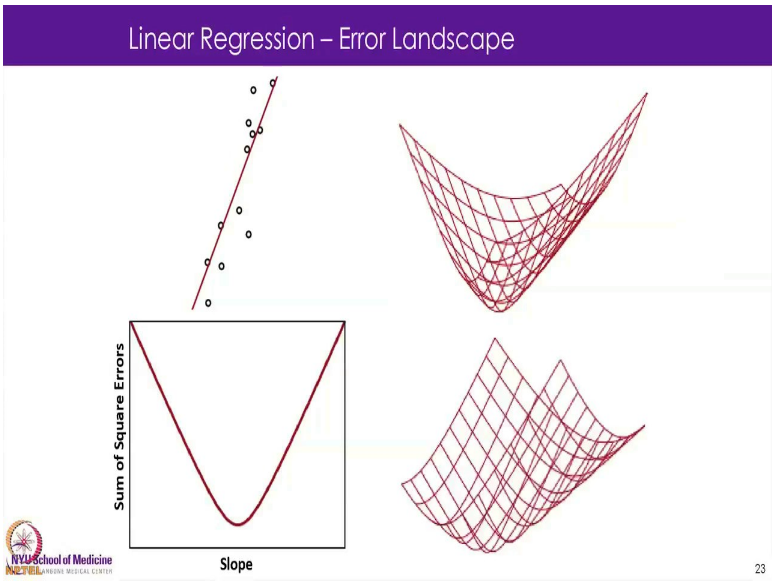
Training: Gradient Descent

Gradient descent can get stuck in local minima.

To improve the behavior for shallow local minima, we can modify gradient descent to take the average of the gradient for the last few steps (similar to momentum and friction).

And, another thing that can happen is that when we get stuck in minima so, that is those are I think the main problems that we run into ok. So, let us look at how this error landscape looks for linear regression.

(Refer Slide Time: 18:07)



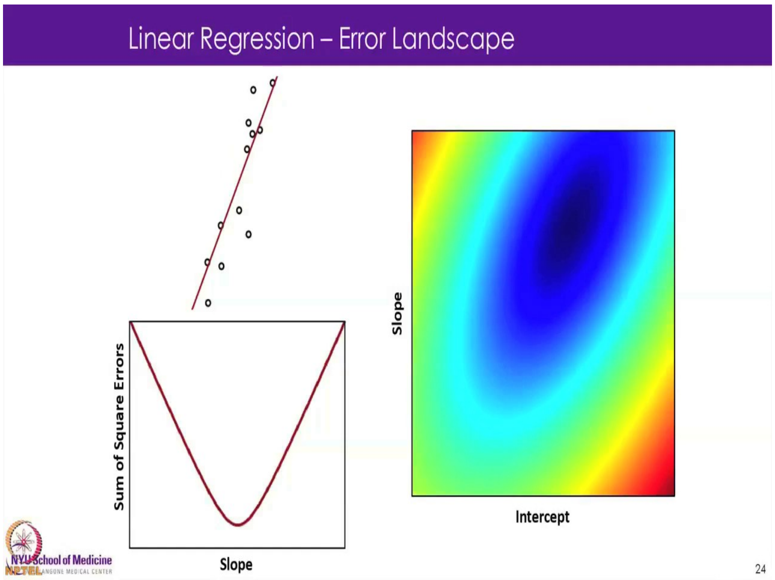
Linear Regression – Error Landscape

Now, as you probably remember from undergrad for linear regression we do not need to do gradient descent because we can actually solve this analytically and we, but we still going to a since it is such a simple case I still wanted to walk you through how it looks if we would need to do gradient descent with linear regression.

So, again so, we have a few points. Here now we have the slope of this and the intercept. So, we have to just look at the slope, if you change the slope from this is the optimal position this is where we have the minimum that is mention. It will the energy the sum of the square errors will be increase.

And, we can look at this in different ways. So, this is the 3-dimensional if you look both the slope and intercept and that rotated.

(Refer Slide Time: 19:11)

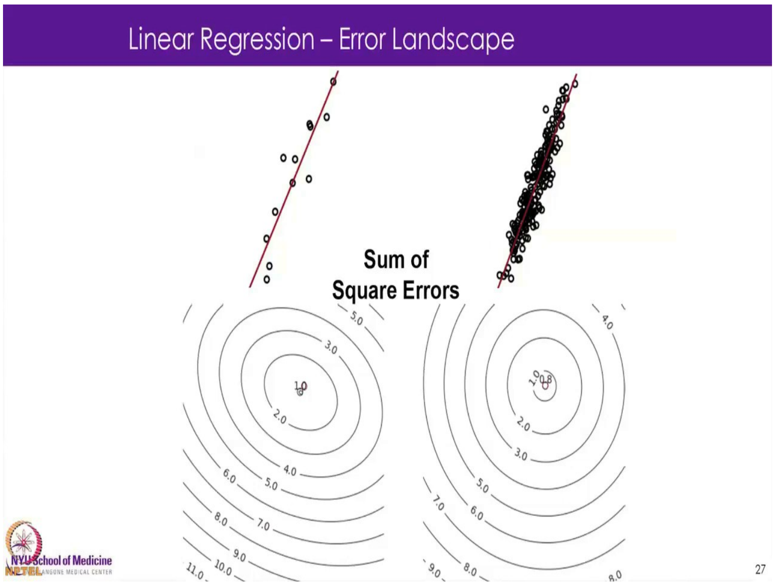


Linear Regression – Error Landscape

We can look at it from above where we have 2-dimensions intercept is here slope is here that is the optimal place where we have the best solution for the linear regression.

And, we can look at it is like this like a map showing the minimum here and the gradient.

(Refer Slide Time: 19:43)

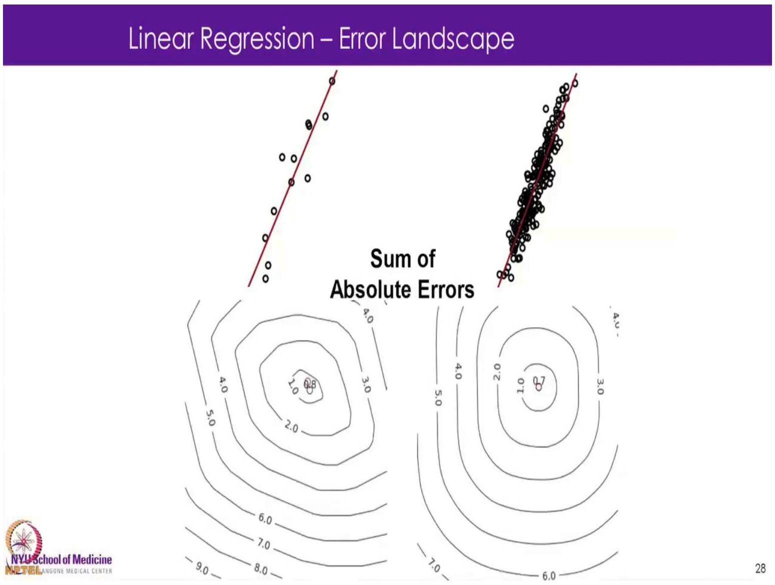


Linear Regression – Error Landscape

Sum of Square Errors

So, if we have two different lines same number of points here and same number of variation we get slight variation in how this energy landscape looks like. But, if we have really a lot of points they the energy landscape is well defined with nice concentric circles which of course, is helpful.

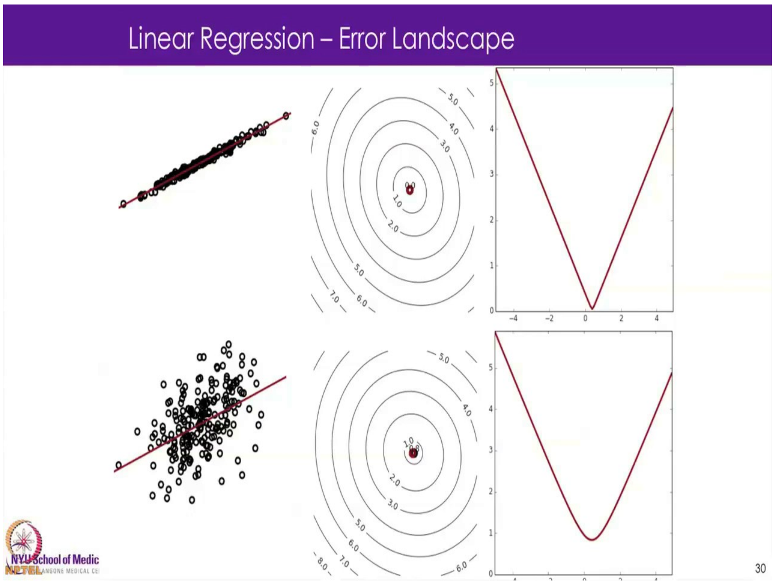
(Refer Slide Time: 20:13)



Linear Regression – Error Landscape

So, we mentioned that it is actually we do not need to use the sum of the square errors, we can also use some of absolute errors. The energy surface becomes a little bit more jagged and not as round, but it is also a possibility and especially when we have outliers that is could be a better solution.

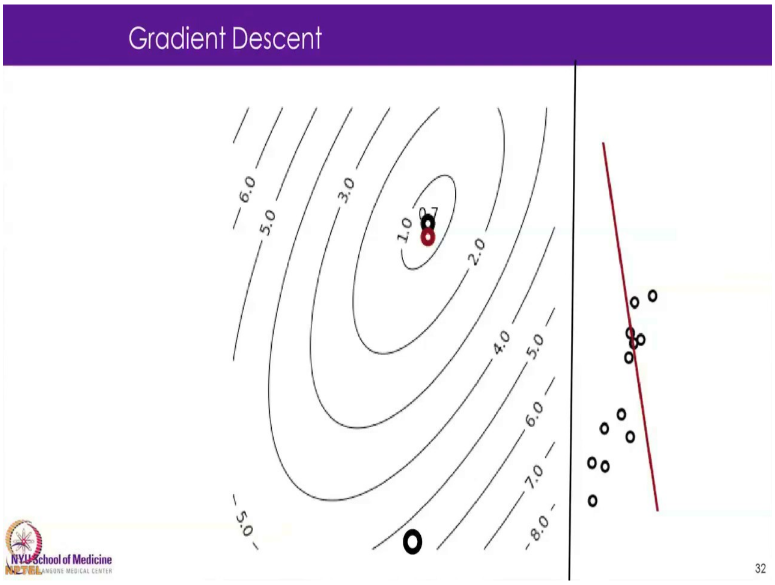
(Refer Slide Time: 20:37)



Linear Regression – Error Landscape

And, so, another thing if we have very little variation so, that is lots of points that define the line well. We get a very sharp minimum, but when we have more error we get a much more much less well defined minima ok.

(Refer Slide Time: 21:03)

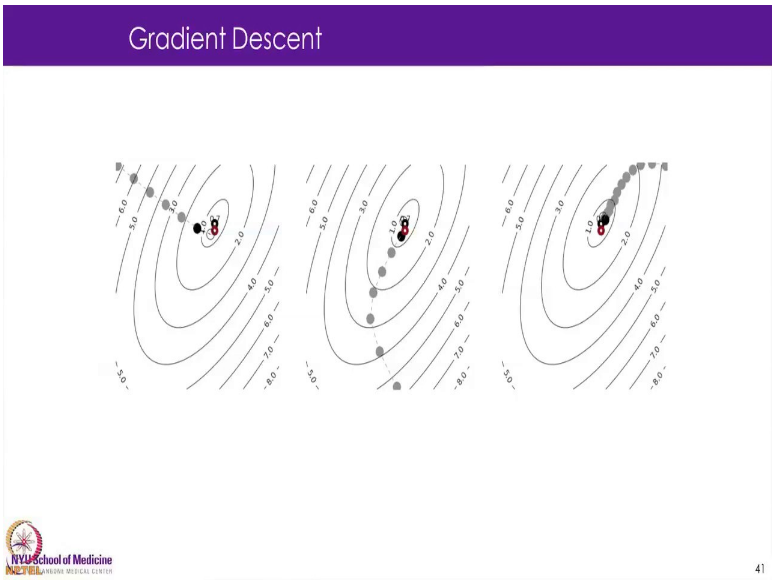


Gradient Descent

So, let us go through a case now we are going to walk down this surface. So, we randomly start here. So, this is again intercept and slope. So, we have our data here this is our randomly assigned line you see that it is not great, but it and it is also because we are pretty far away from the optimal solution.

So, now, we are going to take a small step in the direction downhill at the gradient. So, the gradient is perpendicular to the to these height lines. So, the first step we take would be going perpendicular here and depending on what we choose the step size to be we will take a small step here. And, then we take another step now because of the curvature changes we are going to curve in and then we continue going down following the gradient and eventually and on the side there you see now when we reaches close to the middle the line fits very well.

(Refer Slide Time: 22:19)



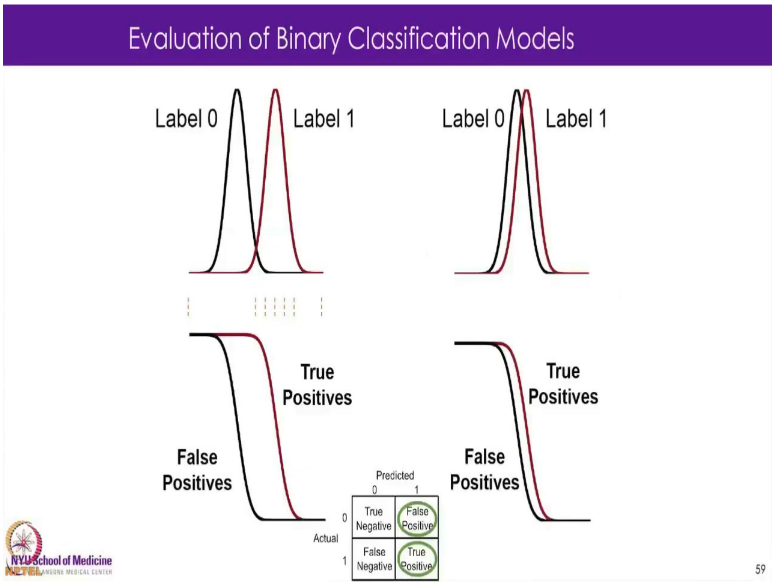
Gradient Descent

NYU School of Medicine
NYU LANGONE MEDICAL CENTER

41

And, and we can randomly start from different places and we end up in the same location because with the linear this is a very nice surface.

(Refer Slide Time: 22:33)



Evaluation of Binary Classification Models

		Predicted	
		0	1
Actual	0	True Negative	False Positive
	1	False Negative	True Positive

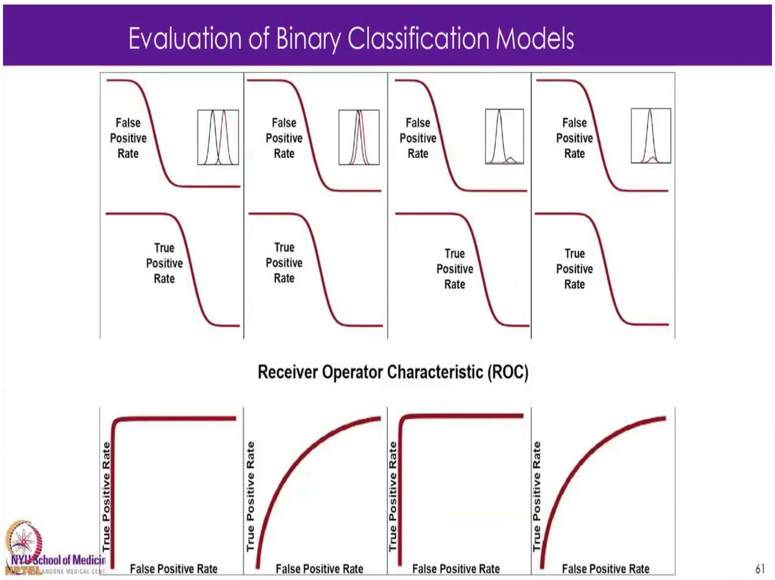
NYU School of Medicine
NYU LANGONE MEDICAL CENTER

59

Some threshold here that gives us a lot of true positives and very few false positives. But, in another case if we have these much closer to each other we cannot do that distinction, but we can still use this to select where to set our threshold.

And, then we can see what happens if there is an uneven distribution.

(Refer Slide Time: 23:03)

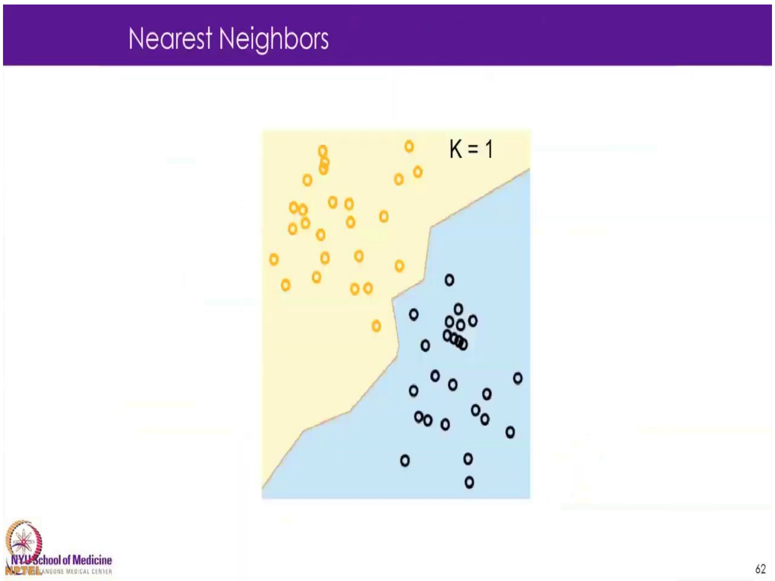


Evaluation of Binary Classification Models

And, so, the other thing we can do is if we have the false positive rate and a true positive rate we can create what is for the receiver operator characteristic an ROC curve. So, how many of you have made ROC curves? So, that is a very common way to evaluate classification, and then when we do comparisons one thing that we in this case we have good separation. So, the ROC curve will start down here and go almost up to the corner here.

So, we have that true positives or separated from the false positives and we can use their for example, the area under this curve to as a characteristic how well we are doing. And, and so, if you have completely random distribution they will be completely overlapping we would just have a line along the diagonal. And, then these are just for a other case where we have done much closer you see that here the curve ROC curve is much closer to the diagonal and these are just cases for the other ok.

(Refer Slide Time: 24:21)


Nearest Neighbors

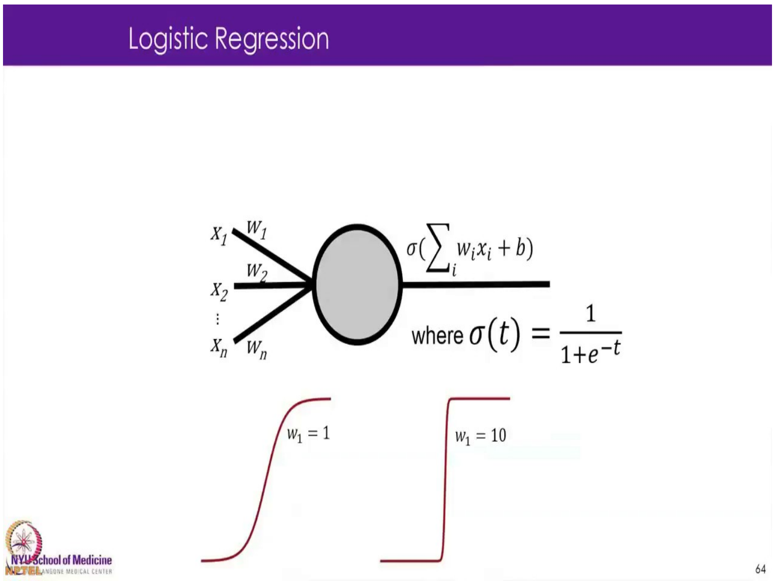
62

So, none of them probably know conceptually easiest way methods is the nearest neighbor. So, here what we do is we just see where what is the nearest neighbor, what is it what class is it is in and of course, if you do evaluate this on the on the training sets you are going to get that error is 0 because it is. So, here is one example of where it is definitely one it shows that why one should not use the training set to valid.

So, here now we have the two groups and the near if you use one nears neighbor we get a good separation between them, but in another case if we take the nearest neighbors when they are more intermingled we see that we get a very complex decision surface and where no one would claim that this is really what it is less. So, this is a very clear case of overfitting.

And, so, now, they can of course, average over a few nearest neighbors in this case two nearest neighbors. Now, it is gets a little bit more plausible, but this for example, there is still an island here of in the middle of the blue. So, then and it gets as we go through more and more nearest neighbors the decision surface becomes more plausible. But, again with nearest neighbors the problem is often that is we have many when we are many dimensions it is just there is nothing that is really near to anything else ok.

(Refer Slide Time: 26:35)



Logistic Regression

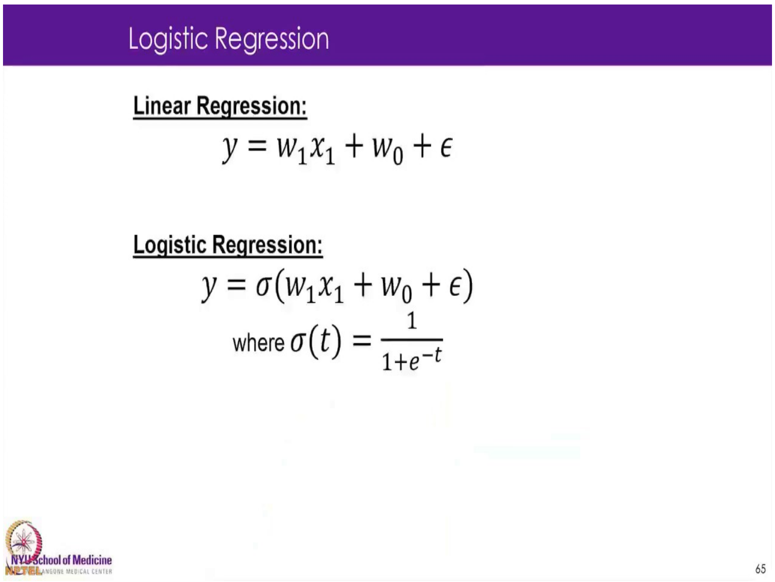
x_1
w_1
w_2
x_2
⋮
x_n
w_n
$\sigma(\sum_i w_i x_i + b)$
where $\sigma(t) = \frac{1}{1+e^{-t}}$
$w_1 = 1$
$w_1 = 10$
NYU School of Medicine
NPTEL LANGONE MEDICAL CENTER
64

So, now a method that is often we can start with to just evaluate this logistic regression. As you if you remember it looks very similar to linear regression that is we have the input or different protein measurements we have our parameters, the weights for which the multiplier each value of it the it is weight, add them up and add the constant, but now that is so far that is linear regression. But, now in logistic regression that is becomes the parameter of a function of the logistic function and which we call sigma and it looks like this.

So, we introduce a non-linearity and the as we see here. So, these are four different parameters of we can we gets this transition from 0 to 1 that is depending on the parameters we have different sharpness so, but what we again. So, now, we have we want to have classification. So, we have two cases. So, in the extremes here at low values we have that is doubt with is 0 and at high it is 1 and then we have this transition region.

So, that is why we can use the logistic function for classification.

(Refer Slide Time: 28:09)



Logistic Regression

Linear Regression:

$$y = w_1 x_1 + w_0 + \epsilon$$

Logistic Regression:

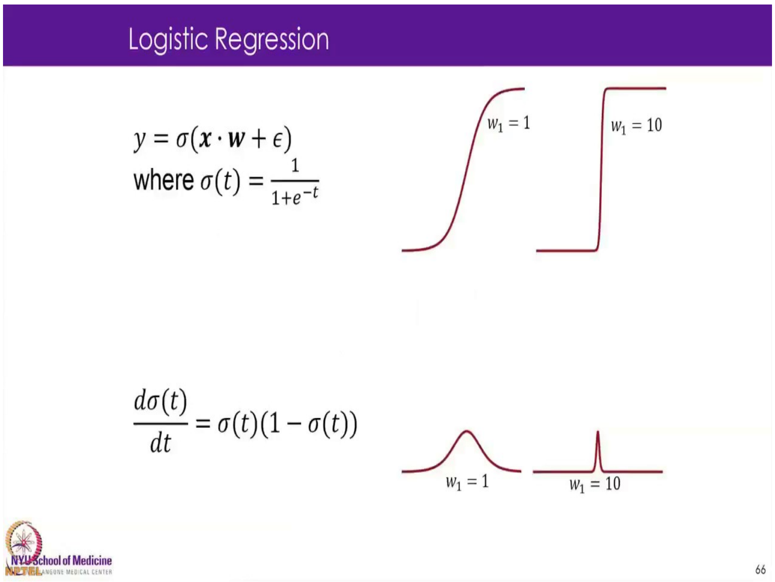
$$y = \sigma(w_1 x_1 + w_0 + \epsilon)$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$

65

And, this now is just comparing them. So, linear regression this is in with one x value. So, we have the slope and intercept that is linear regression and then logistic regression we just have the same expression, but we have a non-linearity.

(Refer Slide Time: 28:29)



Logistic Regression

$$y = \sigma(\boldsymbol{x} \cdot \boldsymbol{w} + \epsilon)$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$

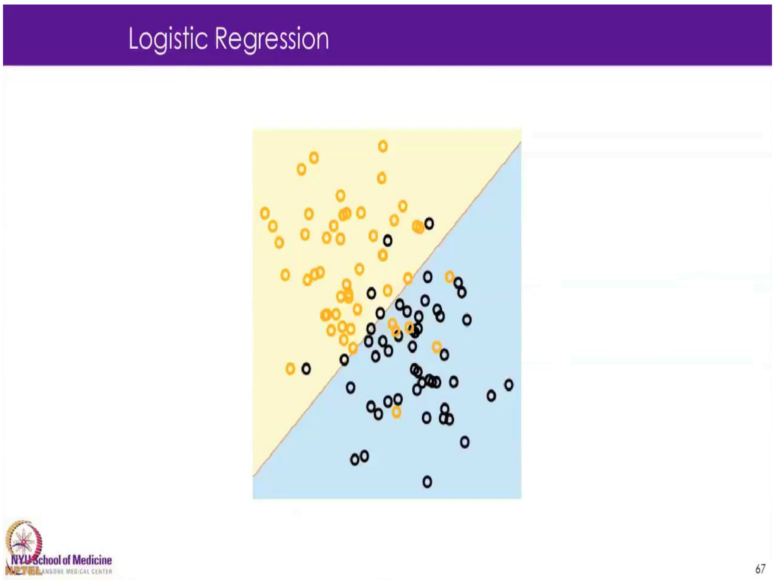
$$\frac{d\sigma(t)}{dt} = \sigma(t)(1 - \sigma(t))$$

66

And, so, the other thing that is we looked at the shape of this function going from 0 to 1 and since we are going to do a gradient descent we need to look at it is derivative and it is there is actually a very simple expression for it is derivative and it looks like this.

So, again it is very flat out here when we follow by from the transition and the derivative is 0 and if you remember for gradient descent that is not very good because we get if the gradient close to 0 we get stuck there. And, so, we want to make sure that we are not too far away when we start otherwise we would not find it.

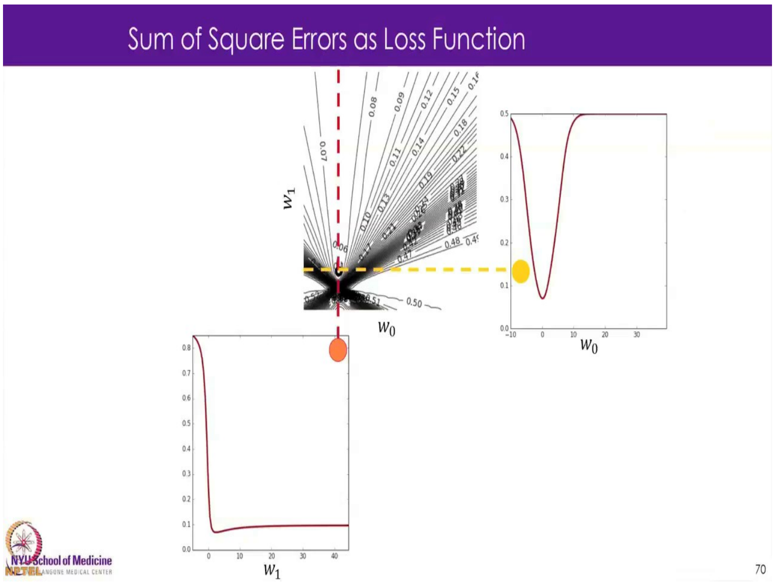
(Refer Slide Time: 29:19)



Logistic Regression

So, this is just an example a very bound logistic regression and if you remember from the nearest neighbor is the same dataset we got pretty close to a straight line that also and that is what we get in this logistic regression.

(Refer Slide Time: 29:41)

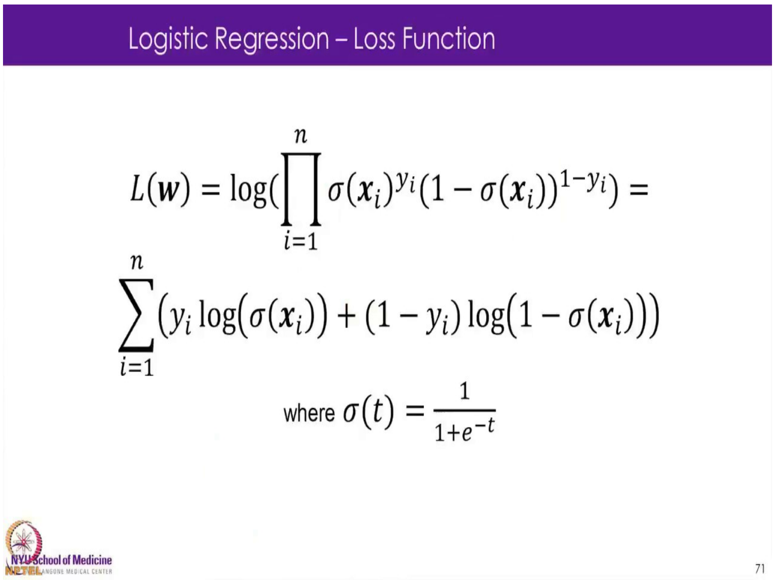


Sum of Square Errors as Loss Function

So, now if we look at the energy surface of this; so, remember that for linear regression when we use the sum of square errors it is behave the really nicely, but here we see something completely different. It is really not does not behave well when you use the sum of squares and it is probably easier to look at it here. So, this is our minimum in there. So, we have a huge mountain behind it, very steep gradients and very shallow gradients. So, we have to somebody find their way in here through very shallow gradients.

So, so, what this means this is a bad choice of a loss function and this is just some other ways to look at it we can have that if we approach from here to the minimum it is very shallow, but then it is sharp and then here we have looking at the other way we have this plateau about where we can also get stuck.

(Refer Slide Time: 30:39)



Logistic Regression – Loss Function

$$L(\boldsymbol{w}) = \log(\prod_{i=1}^{n} \sigma(\boldsymbol{x}_i)^{y_i}(1-\sigma(\boldsymbol{x}_i))^{1-y_i}) =$$

$$\sum_{i=1}^{n}\left(y_i \log\left(\sigma(\boldsymbol{x}_i)\right) + (1-y_i)\log\left(1-\sigma(\boldsymbol{x}_i)\right)\right)$$

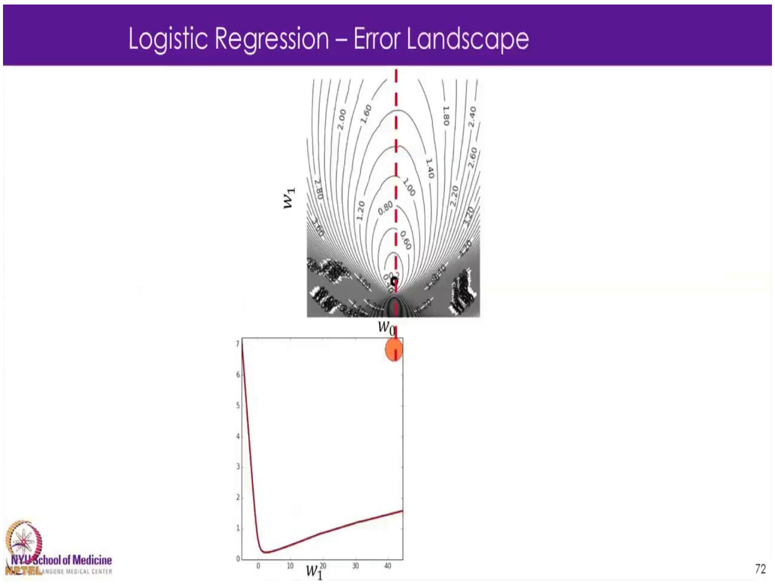
where $\sigma(t) = \frac{1}{1+e^{-t}}$

NYU School of Medicine
NPTEL LANGONE MEDICAL CENTER

71

And, now this is you know me to remember this better is an appropriate loss function for logistic regression and that is this one, but I am not going to go into details.

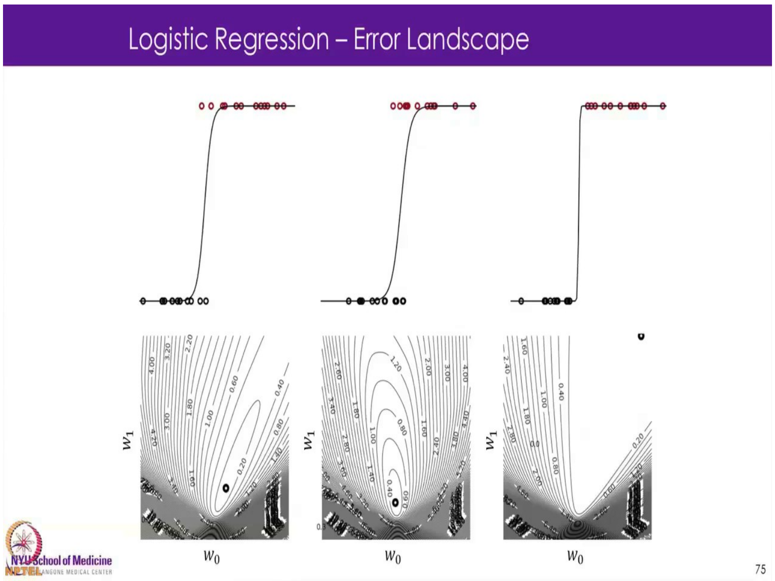
(Refer Slide Time: 30:51)



Logistic Regression – Error Landscape

So, then when we applied that loss function the it is the self has become much more manageable and we can do gradient. We still have that it is in one direction it is more shallow and sharp in that direction. So, it is not as nice of a surface as for linear regression, but it is still reasonably good and also the other thing that for this for logistic regression we do not have an analytical solution. So, here we have to do gradient descent.

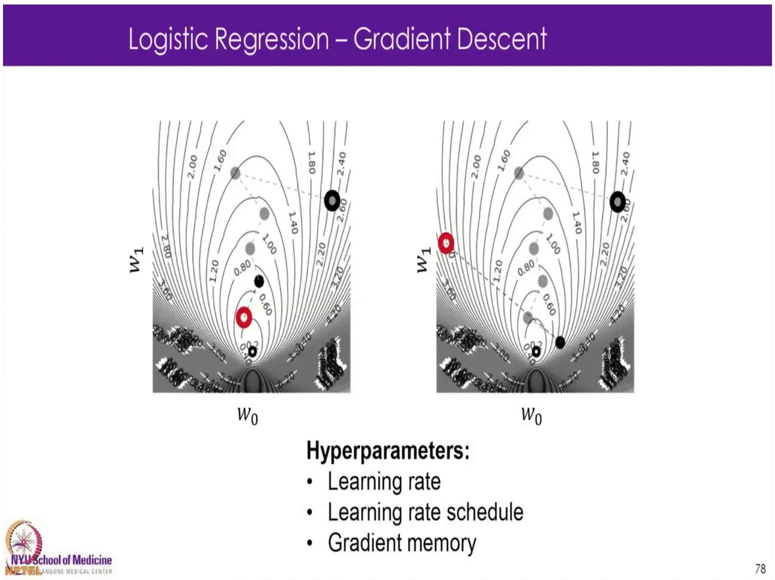
(Refer Slide Time: 31:37)



Logistic Regression – Error Landscape

So, then if you have the same number of points and distribution so, this is one class up here at 1 and then the other class at 0 we see that the surface varies a little bit if we have in the if you have fewer points we get quite a much larger variations in this case.

(Refer Slide Time: 32:03)



Logistic Regression – Gradient Descent

Hyperparameters:

- Learning rate
- Learning rate schedule
- Gradient memory

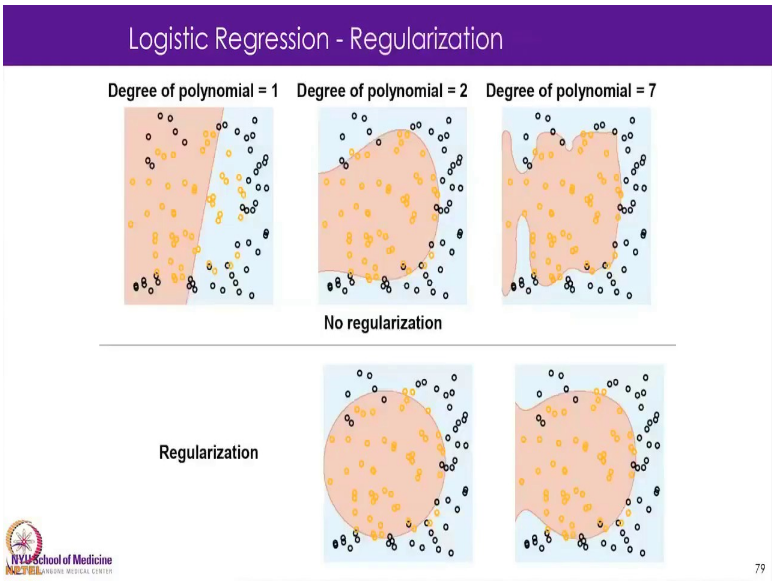
NYU School of Medicine
NPTEL LANGONE MEDICAL CENTER

78

And, when we can also do gradient descent through this; so, we start out here then we walk down, but again we have to be careful that we do not take too large steps when we come close to this very short steep here because then we end up being thrown over far away from the minimum.

So, again so, both for both logistic and linear regression we have these hyperparameter we have to decide on the learning rate, how we scheduled the learning rate usually the how we decrease it and if we want to remember some of the momentum and have some friction built in.

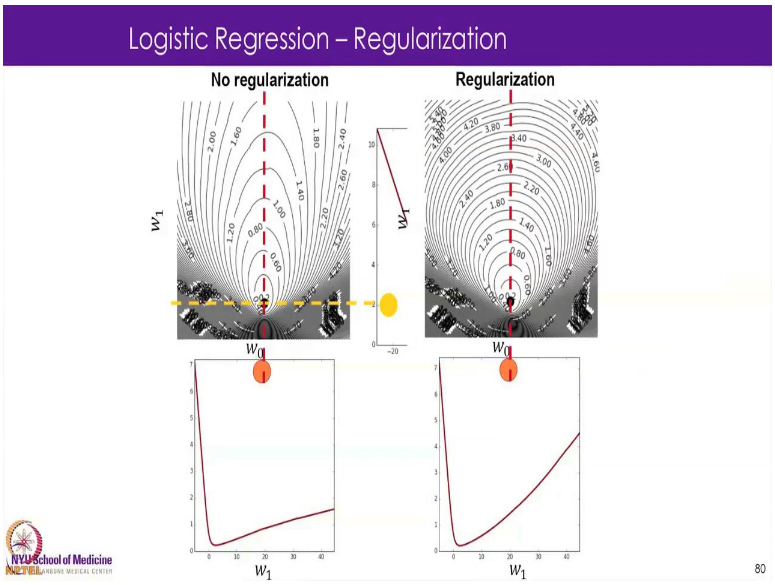
(Refer Slide Time: 32:53)



So, if you look at regularization here again we are on a guard against overfitting, but. So, here in this case So, the same as for linear regression we do not we can also add in polynomial terms. We can do the same thing I mean so, the expression was the same. So, for logistic regression you can do exactly the same thing.

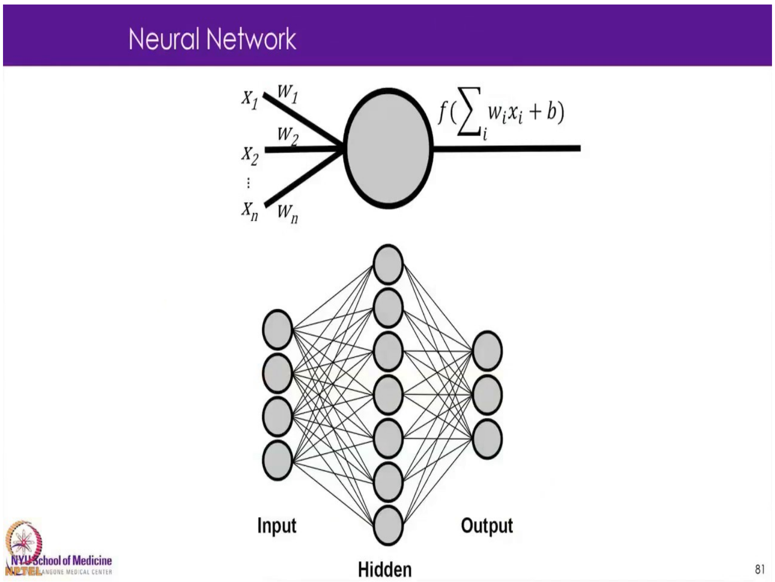
And, so, here we have there is no linear surface that can separate is the yellow and the black here if we have. So, if we add in higher degrees of polynomials they can do a better separation, but again in this case our surface is little bit too jagged and it is probably overfitting, but then we can fix that by doing the same type of regularization either less or ridge regression.

(Refer Slide Time: 33:55)


Logistic Regression – Regularization

So, and how does that if we look at the energy surface for logistic regression with no regularization we had this case and when we at in regularization it actually helps us also in the speed of learning that meaning that and you see that the gradient here is when we add in the regularization is much higher. So, that it is more comparable to this. So, we will be able to find the minimum faster.

(Refer Slide Time: 34:39)


Neural Network

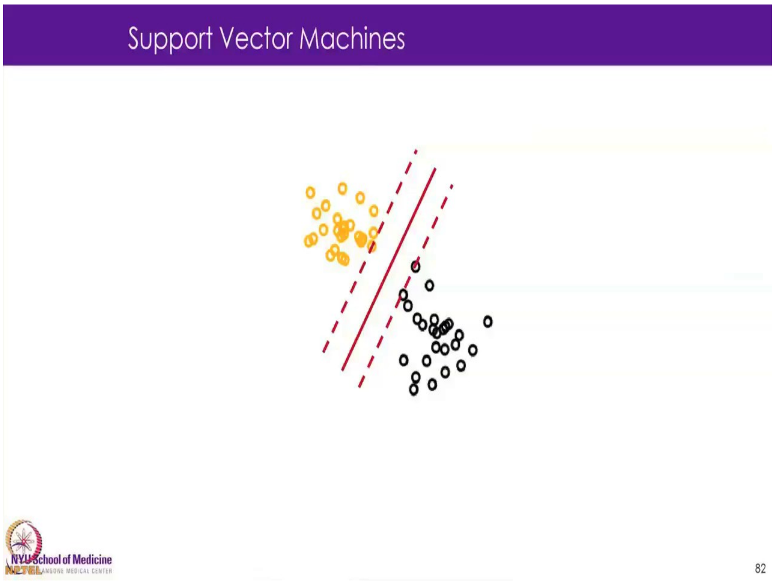
So, then a few examples of yeah So, we had. So, you have probably heard about neural networks and deep learning. So, what that is? So, each of these nodes here is a very similar node to one logistic regression unit. So, we have the inputs the different protein measurements we have the weights and we multiply each weight with each protein measurement and sum them up and then we have an offset.

And, we have some kind of non-linear function which can be a logistic function, but it can also be other things. And, here we have we always have one input layer at least one hidden layer and an output layer and this illustrative what it is called the fully connected network where each node in each layer is connected to all the nodes in the next layer.

And, and now this only shows one hidden layer, but nowadays it is very popular to have many hidden layers and that is what that is why it is called deep learning because you have many layers. And, right now this is the most popular method that probably that people use, but it does often require if you do not want to make it very small neural network, but at least for these large ones that people do you need a lot of data and in most cases for in proteogenomics we do not have enough data to build neural network.

So, most recommendation is even there is all this hype of deep learning , but best to not for proteogenomics not to get into that and unless you have very good reason. And, probably 10 years ago support vector machines were what everyone did and it was a very popular I mean I would say probably support vector machine 10 years ago was what neural networks are now.

(Refer Slide Time: 37:17)



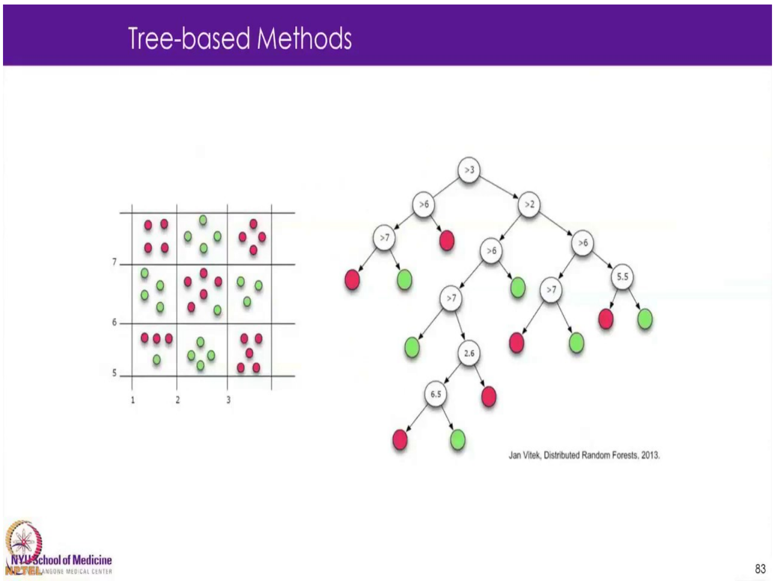
Support Vector Machines

NYU School of Medicine
NYU LANGONE MEDICAL CENTER

82

So, there is always fashion in which methods are you, but support vector machines are very useful and they and most of what they do is they of course, find a plane that separates the data , but then they also find try to find the largest margin. And, the support vectors are the data points that are on these margins.

(Refer Slide Time: 37:47)



Tree-based Methods

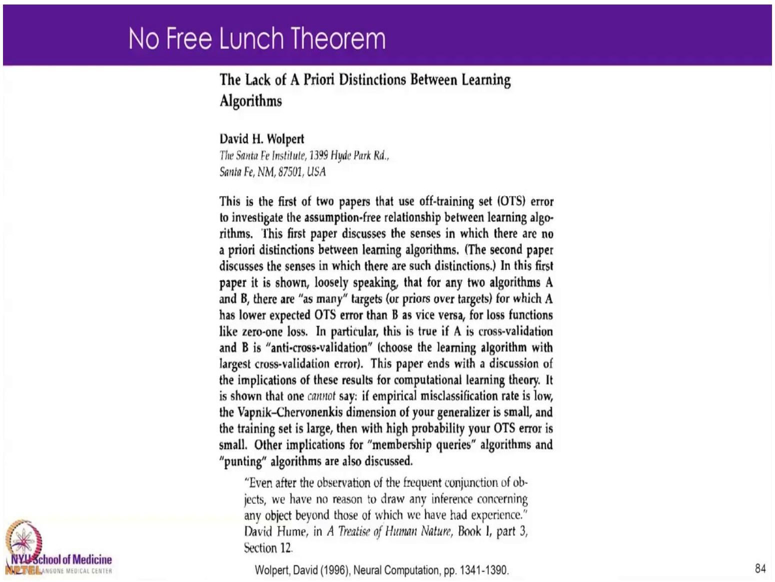
Jan Vitek, Distributed Random Forests, 2013.

83

So, I think Mani showed this slide on tree base methods and those are also very powerful methods that especially in this case showing that you can have a very highly non-linear function that you can classify all these even though they are quite intermingled by. So, each of the nodes in the tree is a decision whether it is some measurement is larger than some or smaller than something, you can go in different errors.

So, I would say that right now the most success people have is with either support vector machines over tree based methods, but actually I would recommend starting with a simple method like logistic regression first also and include those.

(Refer Slide Time: 38:59)



No Free Lunch Theorem

The Lack of A Priori Distinctions Between Learning Algorithms

David H. Wolpert
The Santa Fe Institute, 1399 Hyde Park Rd.,
Santa Fe, NM, 87501, USA

This is the first of two papers that use off-training set (OTS) error to investigate the assumption-free relationship between learning algorithms. This first paper discusses the senses in which there are no a priori distinctions between learning algorithms. (The second paper discusses the senses in which there are such distinctions.) In this first paper it is shown, loosely speaking, that for any two algorithms A and B, there are "as many" targets (or priors over targets) for which A has lower expected OTS error than B as vice versa, for loss functions like zero-one loss. In particular, this is true if A is cross-validation and B is "anti-cross-validation" (choose the learning algorithm with largest cross-validation error). This paper ends with a discussion of the implications of these results for computational learning theory. It is shown that one *cannot* say: if empirical misclassification rate is low, the Vapnik–Chervonenkis dimension of your generalizer is small, and the training set is large, then with high probability your OTS error is small. Other implications for "membership queries" algorithms and "punting" algorithms are also discussed.

> "Even after the observation of the frequent conjunction of objects, we have no reason to draw any inference concerning any object beyond those of which we have had experience." David Hume, in *A Treatise of Human Nature*, Book I, part 3, Section 12.

Wolpert, David (1996), Neural Computation, pp. 1341-1390.

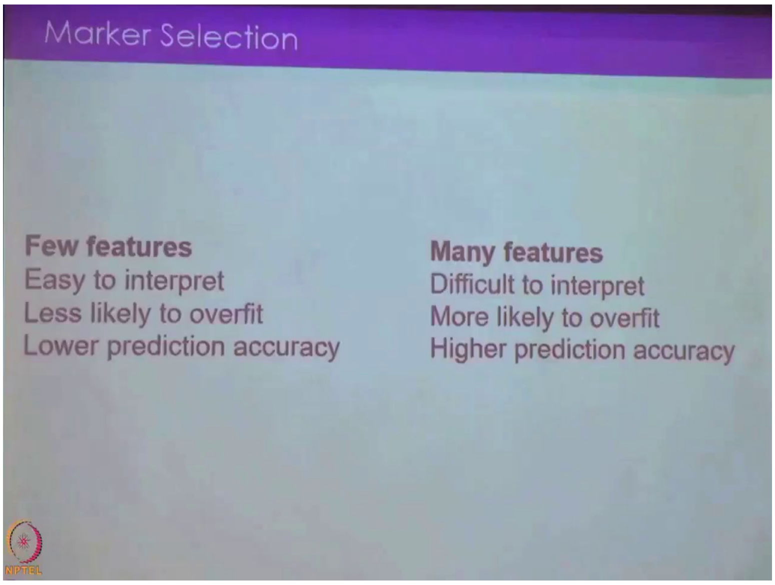
84

And, the there is actually there is a theorem that is called a no free lunch theorem. And, this was in the 9 days some people showed that when you start with a new project you have a new dataset that you do not have experience with there is no way to tell which method we will work best.

So, it is really sometimes a tree based method like random forest will work, but sometimes logistic regression, sometimes support vector machines. So, it is really and of course, all the methods have lots of parameters that need to be adjusted. So, what people often do is they try all possible methods.

Now, of course, what you cannot do is to train one method on your training dataset, test it on the test dataset, train another method or you are training to test it also and do this many times for all the both for a different methods and for a different hyper parameters because you should only use your test set once. So, you need to do this exploration, you need to do within cross validation and they I think they are almost getting to cross validation I have said.

(Refer Slide Time: 40:27)



Marker Selection

Few features	Many features
Easy to interpret	Difficult to interpret
Less likely to overfit	More likely to overfit
Lower prediction accuracy	Higher prediction accuracy

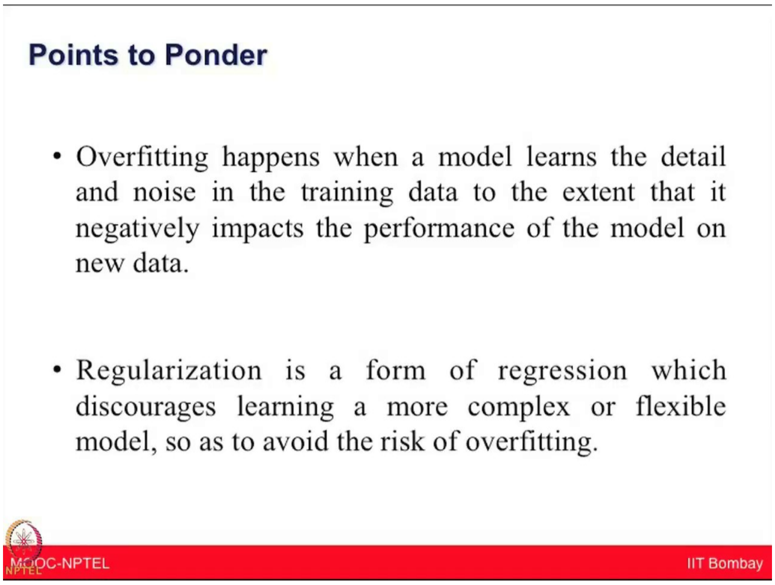
That is we are going to get there soon, but and so, the other thing is marker selection that is we have already mentioned earlier. So, now we do all these measurements and we you we know that most of the proteins or most of transcripts are not going to be related to our phenotype. So, we really it would be much better to just have build the model using the ones that we know are related, but of course, we do not know which ones to start with. So, we need to find.

So, they are if we look at mark, so, by the way do marker selection. So, the having few features it is makes the model easier to interpret. So, one thing that we have talked about building these predictive models and we want to predict something, but if we can also understand that is of course, a much better thing and off the many build very complex models we do not understand and maybe would not have a chance to understand.

And, few features so, it is easier to interpret we can start thinking about biological function and they are also less likely to over fit because fewer parameters, but usually we get a little bit lower prediction accuracy. So, that is something to balance and that is what we use to the then decide how many features.

So, as a person so, if there are many features it is difficult to interpret we do not know what is going on and then of course, more likely to over fit because we have do have an enormous amount of parameters. But, of course, as we add in more and more things we gets higher prediction accuracy, but it is we are not sure whether that is really real.

(Refer Slide Time: 42:31)

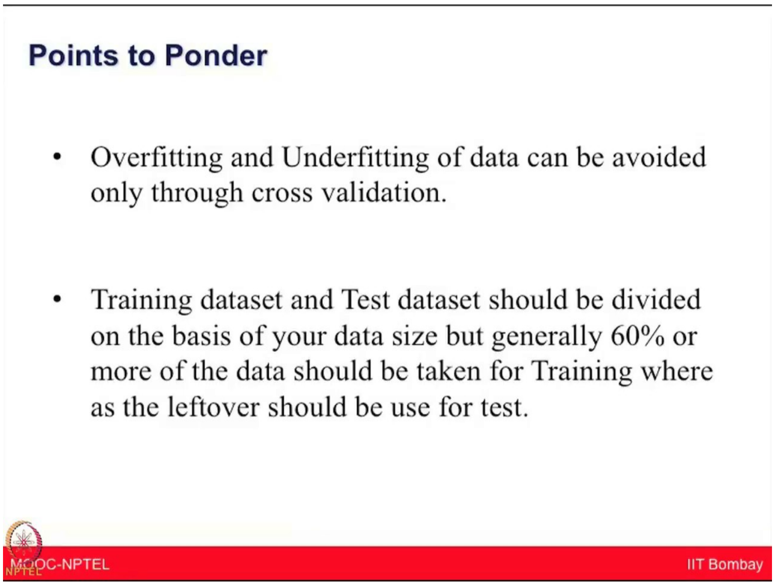


Points to Ponder

- Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.
- Regularization is a form of regression which discourages learning a more complex or flexible model, so as to avoid the risk of overfitting.

MOOC-NPTEL IIT Bombay

(Refer Slide Time: 42:43)



Points to Ponder

- Overfitting and Underfitting of data can be avoided only through cross validation.
- Training dataset and Test dataset should be divided on the basis of your data size but generally 60% or more of the data should be taken for Training where as the leftover should be use for test.

MOOC-NPTEL IIT Bombay

Dr. Fenyo provided a very good overview about how separating your dataset in different training or test models can give a better evaluation. We also learned that when there is a increase in the degree of polynomial the error goes down. We also learned it is better to have large dataset as it will help in evaluation of the model better. Finally, we understood how to minimize the risk of overfitting of data with regularization and why we should avoid overfitting of data. We also understood two regularization strategies which can be used like Ridge and Lasso.

In the next lecture, Dr. Fenyo will talk about Association and Marker Selection.

Thank you.