Artificial Intelligence for Economics

Prof. Adway Mitra

Artificial Intelligence

Indian Institute of Technology Kharagpur

Week – 03

Lecture - 12

Lecture 12 : Linear Regression and Classifiers

Hello everyone, good morning and welcome to this course on Artificial Intelligence for Economics. I am Adway Mitra, an Assistant Professor at Indian Institute of Technology, Kharagpur and today we are going to discuss about Linear Regression and Classifiers. So, we are currently on lecture 12 of this course in the like we are right now discussing with some machine learning models and algorithms for using which we can extract knowledge from past data which we have and use them to make predictions on future data. So, like we have already discussed the concepts of like unsupervised learning or clustering where we saw k means clustering or agglomerative clustering and in the previous lecture we also discussed a simple classification algorithm which is known as the decision trees. In today's class also we will discuss more on supervised learning and we will like specifically talk about linear regression and linear separability and and classification and we will see some algorithms for both of them. So, in case of supervised learning we have all like as we have already discussed we are provided with observations along with their levels.

So, the labels are like $y_1, y_2, \ldots, y_n$ and the each of the observations is represented by a feature vector $x_1, x_2, \ldots, x_n$ where each of the like these x_1 s. So, n in this case is the number of observations for each observation I have a vector called x which which might be d dimensional. So, each of these I mean each element of this vector may be some attribute which is like which we can calculate in some way. So, we say that in general the feature vector it belongs to some space which we may call as the feature space which may be one of the standard spaces we deal with in mathematics or it might be some like unusual or abstract or mixed space also.

The label vector also similarly belongs to the label space ok. So, this y_1, y_2 all of them we say they belong to the label space. Now, what is the aim of supervised learning? The aim is given a feature vector you have to predict it label. So, which basically means we need to learn we need to find a function which maps the feature space into the label space

in the in this way $y=f(x)$ that is x can be any arbitrary point in the feature space there should be a function f that is if we apply that on x then we get a value in the label space and that is what we can call as the y. this function f it can be an explicit mathematical formula function like say some polynomial function or trigonometric function or something or it can be like that is an we can say that is an explicit function or it can be an like an algorithm like for example, decision tree.

In case of decision tree you are taking in a feature vector and you are predicting the level, but you are not you do not really have a specific mathematical function by which you can express what the level will be in terms of the feature vector. But instead you have an algorithm you just know how what to do with the feature all the elements of the feature vector and come up with like an answer which is like a value in the level space. So, it can be either so, this f it can be either an explicit mathematical function or an implicit mathematical function which is expressed with the help of an algorithm ok. So, now, if this label space if that is discrete then we call this problem as classification, but if the label space is continuous or real value then we call it as a regression problem. Now, the what is the general approach for supervised learning? So, we first make a hypothesis about the function f for like.

So, like as I said it can be either an explicit mathematical function or an algorithm let us say it is an in this case it is an explicit mathematical function. So, in this case we just assume some kind of a form for that function like one example is written there I mean this is just for showing. I like in that case I may have reasons to think that it can be a quadratic function like this or it may be some other kind of function. So, the you just make a hypothesis about the type of the function. Now, we need to calibrate the function with the observation.

So, note that we are provided with these n observations for which I know both the feature x as well as the label y that is the input and the output are both known to me. So, I now it is expected that the I have to find the function f for which this relation holds that is I should it should that is whatever f I am coming up with it should satisfy that y 1 should be equal to $f(x_1)$ or $f(x_2)$ should be equal to y_2 and so on and so forth. So, how do I come up with such an f. So, the form of the f I make an hypothesis about, but then apart from that there are the parameters like this a b these are not things these are like that is my I have hypothesized that there will be a parameter like this, there will be a parameter like this and so on, but what their exact values will be that I have not hypothesized. So, I will have to try to calibrate the model with the observations which we have got so that I can estimate what the values of these a and b are.

So, calculating the values of these parameters in whatever function we are trying to fit

that is known as the calibration or training and whenever we want to do calibration we must need some reference. So, the reference in this case comes from the level data that is for which I know both the input x and the output y. So, that is that also known as in the jargon of machine learning this data provided for calibration is known as the training data which is of the form (x, y) it is commonly written as a tuple like this the first is the feature vector or input and the this is the label or the output. So, now how do we do the this calibration? So, the general idea is that like whatever value or whatever function f you are choosing including the values of its parameters it should be able to make successful predictions at least on this reference or training data set. That is whatever function f I have chosen if I apply it on x_1 I should get y_1, if I apply it on x_2 I should get y_2, if I apply it on x_n I should get y_n, but is that happening or not.

So, like the function will make a prediction that is called as $f(x)$ now I have to compare that $f(x)$ with the true level which is y. So, I need a loss function which measures the discrepancy between the true and the predicted values that is in this way L which is the L is a binary function it takes two values as inputs the one is the true level y and the other is the predicted level y right. Now, if we have n such examples then I can actually calculate the loss over all of these examples like this. So, now, what is the value this is the training phase and then once you have calibrated the model that is by what is your task is when doing the calibration is of course, to minimize this loss. Ideally y should be exactly equal to $f(x)$ that is in that if that is the case then the loss function should may have the value of 0.

y_2 should be equal to $f(x_2)$ in which case the calibration I mean the loss function in this case will be equal to 0 and so on and so forth. In practice loss function getting it to exact 0 may be very difficult, but at least it should be minimized as low as low as possible. So, now once you have chosen the function f for which the total loss over all the training examples is as low as possible, then you have to validate the model. So, now you apply it on any other example which was not part of your training process and see whether it is able to do a successful prediction on that also. So, that is the validation phase.

So, let us take a very simple example of classification. So, let us say this is a toy problem the cat dog prediction problem. So, you are given an image based on which you have to predict whether it is a cat or a dog. So, in this case you have like the for training purposes you have images of cat and you also have images of dog each of them are each image is represented by d number of features including their values. So, both of the all these features of the different classes along with their class levels are provided as inputs to your learning algorithm.

What is the learning algorithm? The learning algorithm is what will is the process using

which that loss function will be minimized and the values of the parameters will be provided to you as output. The form of the function f that is which is your hypothesis that is built into the learning algorithm or rather we can say the learning algorithm will be specific to your choice of this hypothesis. But once you have made the hypothesis the learning algorithm will take as input these kinds of level data and provide you the optimal set of parameters which minimize the loss function at least with respect to those examples. Now, how it will do with respect to some other example we do not know. So, for that we have the testing or the validation phase.

So, we now test it with an example which was outside the your training phase. So, that testing example again has its own features you provide it to the learning algorithm which has been trained. Now based on this your model makes a prediction that its level is dog which while the its correct level is of course, cat. So, here we detect that a misclassification has taken place that is its predicted level is dog while its actual level is cat.

So, there is a loss. So, immediately your learning algorithm will raise an alert. So, again you will have to go back to the original set of examples used for training and make appropriate changes to it that is the value which means the parameter values which were being used. So, far they are that means, it is not good enough because they are making mistakes. So, somehow the learning algorithm will update those parameter values and send back the new parameter values. And once you apply the new parameter values then you may be will find that the predicted level is cat while the true level is also cat.

So, they like now there is no loss. So, we can say that the your training is successful. Now, we can like base. So, this is the broad idea of course, of supervised learning. So, now, based on super as we say discussed earlier supervised learning has two part has like two problems one is regression and the other is classification.

So, let us consider the regression problem first. In fact, let us consider the simplest kind of regression. So, what is simple about it the simplest assumption we have made the simplest assumption about f what is f the function which maps from the feature space to the label space and what is that simple assumption the it is that it is a linear function ok. So, that is like we can write it in this way w transpose $x + w_0$. So, x as I said is a vector.

So, if I w is another vector of coefficients. So, if I write it like it will be w transpose x plus w naught this extra term with w naught that is as a bias term it can be absorbed into the weight vector. So, that I can just drop it and just write it as $f(x) = w^{tr} x$. So, now, the task is to estimate this weight vectors. Now, if this were like if the instead of feature vector if it was just one dimensional features it would be equivalent to just fitting all the

observations on a single straight line that is the kind of regression which we had all studied in the in our 12-th classes.

If we are it is we are dealing with a higher dimensions instead of one dimension then instead of a line we are trying to fit them into a d dimensional plane which is also known as a hyper plane. and the loss function which we can use in this case say that is your this predicted value will be some real number obtained in this way your actual value is I mean the true value is also a real number. So, you can just calculate the difference between them and square it up. Now, why am I squaring up? So, that like there is no the we can deal with the problem of the positives errors and the negative errors canceling out each other. So, that in the like if I put the squares with the all the losses will always be positive.

So, they cannot like they will only add up if you make mistakes they will add up, but if you did not have the squares. then a positive mistake and a negative mistake would have cancelled each other. And although you are making two mistakes it is possible that your loss function would still show 0. So, to avoid that predicament we are using the squared error loss function. Now, why do I care what can be possible application of this task? So, let us consider the task of how customers rate a product.

So, once in the previous case of decision tree we are talking about ratings rating prediction in this case also we are talking we will discuss the same thing. So, let us say that the some which some service or some product has been given to many users. and they give ratings to different aspects of that based on which they come up with overall rating for the for those products or those services ok. So, what is your aim? Your aim is given the different features of the service or the product you have to predict its like the how much rating the user will give to it. So, we consider that every feature product has so many different features.

Now, the a rating y i like can be considered as a weighted average of the scores which the user may be implicitly giving to the different features. So, like in some cases we like if you look at this example here the user is giving ratings to individual features of the service. On the other hand in many cases they may give an overall rating to a particular service. So, like we can imagine it to the overall rating to be a some kind of a weighted function of the individual feature ratings. So, now, the that each of the features will have different weightage will have different importance to the different users.

So, some features as in case of decision tree also we discussed that some features may be important in determining the overall rating, others might be less important. So, this in this case like in case of decision tree that was quantified through entropy and information gain, in this case it will be quantified through the values of these weights w's. So, we can

say that rather than considering individual scores or the scores given by the user to the individual features, we can consider them that overall rating y is a function of the values of the different features themselves. With the assumption that higher the value is of a feature is the higher should be the users satisfaction with that particular feature. In general that is not the case, but we can like modify it to make that the case.

So, that we can write the final rating y is a linear combination of the like of the values of the different features weighted by these coefficients like this w_{ij}. Now, what is w_{ij}? w_{ij} is basically the importance which the i-th user applies to the or has for the j-th feature. and what is like similarly what is x like x_{ij} is something like the score which the i-th user has given to the jth feature which we consider is just a function of the value of the j-th feature itself. So, like there is a I think I there is a mistake here these the feature values we will not consider as $x_{i1}x_{i2}$ etcetera these are independent of i that is these are not i is a user.

So, these will be $x_1, x_2, \ldots, x_n$. So, this is a typo here which I will correct. So, anyways so, here what we do we need to do we need to estimate these weights w. Now, there are m which are users and n features. So, total m n number of these w's are there because different users will have different weightages for the different features, but estimating so, many parameters is generally a difficult task. So, now I do not want so many features in my model like as a general principle whenever I am building any model or any function f I will prefer such functions which have as few parameters as possible.

Having too many parameters is always a dicey issue because each of them have to be estimated which is often a difficult task. So, I want to simplify the model and reduce the number of parameters. So, what is my new approximate model? The approximate model is that like different all the different customers they have the same weightage for up any particular feature that is instead of considering w_{ij} I will just consider w_j where w_j is like the common importance for the j-th features that is all the features will have equal weights for each of the features. So, this is the new model.

So, we can write it as $w^{tr}x_i$. What is so, like if we are considering the x_i as the like the features for the for one particular product. then to for that I and now if I want to predict how much rating any user will give to that product. So, I can write it as according to my model that can be calculated as $w^{tr}x_i$ right and i now no longer means the customer, but it means any particular object. or any particular item. So, the predicted rating is h_i, but the true rating is like is y_i.

So, we need the loss function as we decided earlier the squared error loss function. So, I compare y_i with this $w^{tr}x_i$ which is just which can also be written in this particular way

that is I can write it as the summation sorry one term here is missing there will be a summation over j. So, just pardon my error this summation is over i there will be an internal summation over the j's also I will make a correction to the slide here. So, anyway so my task he will be to choose w to minimize this total loss over all the users and then differentiate the total loss with respect to w equate that to 0 and then solve it. So, like as I said earlier also the loss function can be calculated over every individual observation or and then we can sum up the loss function over all the observations.

So, that is the total loss now I need to choose the parameter values w. So, as the total loss is minimized ok and so, how do I do that I just take the I just sum up the all the losses that I mean to calculate the total loss differentiate it with respect to w and equate it to 0 and just solve it. So, it turns out that the w you can calculate it in this particular way w turns out to be $x^{tr} x$ whole inverse times $x^{tr} y$. So, a what is a like x, x is a matrix which is formed by this one like this $x_1, x_2, \ldots, x_n$ and y is also by $y_1, y_2, \ldots, y_n$. So, like all like like all the observations which we have we just like like for each of the observations we have a vector and m dimensional vector which we just like line up one to next to each other to get an m cross n matrix like this.

So, that forms your x. So, you calculate $x^{tr} x$ which is also which is of course, an turns out to be an $n \times n$ matrix you take the inverse of it and you do this $x^{tr} y$ which is another matrix multiplication. So, that you finally, get an n dimensional vector w. So, what is n? n is the number of features. So, for each of the features you have calculated their weights.

So, that is the how you do the linear regression. So, now, the the important thing to remember is that this is this is what your regression model is. Now, you have got the like the weightage of all the individual features using which you can calculate the overall ratings. But the assumption is that in the all the feature ratings they are contributing to the final rating. But in reality usually when I as a customer give a rating to any article I will not consider all the features to it only some features will be useful for me and the others I will not even consider. So, it is like so that means, that maybe only some elements of w will have high values or the other elements will have 0 values.

or it may be that some of the or maybe a few elements will have 0 value and only 1 or 2 may be may be having non 0 values that is we will say that w is pars. So, now, we may have that we may consider the task of feature selection that is the task of identifying the important features that is those features which have which should have high values of w. I mean to say anyway the like we can solve the linear regression problem and then see which features are having high values and which are having low values. But suppose I want to explicitly or actively look for discriminate between the important features and the less important features. That is I want the important features to stand out with high values

and the non important features to disappear with values close to of w which are close to 0.

So, we like so, that is what we want. So, these demands can be convert that into mathematical formulation. So, it turns out that we can. So, we can add something known as a regularizing function called $f(w)$. So, now, the so, the first is the loss function which makes sure that your predictions are as good as possible that is given any x you are able to predict the correct value of y. while $f(w)$ is the regularizer which means that the w vector or the set of parameters which you are getting they satisfy whatever properties which you want.

So, one such property as we already mentioned can be sparsity. So, like we can if we choose f w to be the l 0 norm of the w vector it can be shown that like you you will get a sparse solution of w. But, the problem is how do we carry out this optimization. So, earlier in a some of our previous lectures we have discussed about optimization algorithms. So, like we have discussed about interior point, exterior point methods and things like that.

So, it now it the problem is if this function $f(w)$ it becomes non continuous and if we choose this particular f and in that case it becomes very difficult to make a prediction. So, to make the correct optimization. So, the relaxation so, we want to make a relaxation instead of this 0 vector sorry this l_0 norm we consider the l_1 norm which it will give us the almost sparse solution. That is like if we had minimize use this as the regularization function it would have given us the sparse solution, but solving the problem turns out to be difficult. So, I use a different function which is a relaxation it will give me almost sparse solutions.

So, the so this problem can be this optimization problem can be solved using some of the methods which we had discussed earlier. So, this is known as lasso regression. Lasso stands for least absolute shrinkage and selection operator. Now, the like if you consider like remember this parameter λ. So, as I said we have two objective here one is the prediction the other is imposition of a certain kind of structure like sparsity on the W.

So, this λ it in like indicates which of these two tasks you are giving a higher value. So, as we you can see that as we are increasing the value of λ the loss function gradually increases while the sparsity of the w vector that also increases. The loss function increasing means that you are making more mistakes in the prediction, but at the same time you are making like you are getting the sparse kinds of w which you are interested in. So, it is really a kind of trade off between the two objectives which represented by the w. Now, so far we have been talking about the task of regression, but we can maybe we

can also do classification using the like this linear using the linear function.

So, in this case the our the level space has only two values we specifically talk about binary classification. So,from the binary if we can do binary classification it can be shown that we can do any kind of classification also even if the number of classes is more than that it can be broken down into a binary classification into a number of binary classification problems. So, if we can solve the binary classification problem, so we can solve the entire classification problem. Now, the binary levels are let us consider are plus 1 and minus 1. So, what we are looking for basically is we have two sets of points or ok no we have a set of points which we are trying to partition into two sets the first one having levels plus 1 and the others having level minus 1.

So, how can I do this partitioning? the like if we are like just for visualization if we consider this 2D feature space when there are only 2 features that is x is a 2 dimensional vector. So, I can try to like divide the 2 types of points by drawing a line and my assumption is that all the points on one side of the line will should belong to one class that is one value of the y let us say minus 1. and the all the points on the other side of the line they will have a different value of the y that means, it will belong to a different class let us say for the plus 1. So, the so, this is my task. So, I write it the model I write it in this way y_i the predicted value is equal to sign of $w^{tr} x_i$.

So, if I calculate this quantity $w^{tr} x_i$ it is a real number. So, it can be either positive or negative. So, if it like if that is we can say in general that if x_i lies on the plane w on the hyper plane w then this $w^{tr} x_i$ will be equal to 0 if it lies on one side of it it will be positive if it lies on another side of it it will be negative. So, accordingly we like we whatever the value of $w^{tr} x_i$ is I apply the sine function to it and accordingly I make the prediction about y. So, in this case since it is classification the loss function is 0 1 which is basically just the count the number of times the predicted and the true levels do not match. So, now like if there exists a linear classifier like this which achieves 100 percent accuracy that is it is able to there exists a linear structure which is a line on 2D and a hyper plane in higher dimensional higher dimensions such that all the points of one class it is it has on one side of it and all the points of the other side it has or the other class it has on the other side then we say that the data set is linearly separable not all data sets are linearly separable in fact most are not.

But, if there if it is linearly separable then there exists a linear classifier which will achieve 100 percent accuracy. So, the task is then how do we estimate w from the training examples. So, there is one really simple algorithm which is known as perceptron it starts with an any initial value of the coefficients w. It tries to classify all the examples

one by one that is it applies that prediction sign of $w^{tr} x_i$ and it compares the predicted value with the its true value which is y. Whenever it makes some mistake it corrects itself that is the value of w is changed according to a simple relation and if we do this over and over again we will finally, find the value of w if one exists that is if the data set is linearly separable.

But will that be the it will be some value of w which will classify the examples perfectly, but will it be the best one. So, let us say that there are two possible solutions you see that there are the two types of points two classes of points the blue points and the red points. And in both of these cases we have got a valid solution a linear classifier which is able to classify them correctly. But these two solutions are not equivalent. So, we can say that in case of solution 1 the classifier is very close to this point or to this point.

So, if there is a small amount of noise in the feature then in the feature space this point will move a little bit to this side or this point will move a little bit to this side which means that they will be misclassified. But, in this case that problem is not there. So, we prefer such a linear classifier which is as far away from the examples as possible that is called as a max margin linear classifier. So, such a max margin linear classifier can be estimated using an algorithm which is known as support vector machine which requires complex optimization and quadratic programming. We will not go into the details of support vector machine how it works, but let us just know that it allows us to find such a linear classifier which maximizes the its merging from all the classes.

So, we discussed binary classification, but it can be shown that like if we can do binary classification, we can also do classification into any number of classes. So, what is the applications of these in economics? So, both regression and classification they are useful in various tasks of economics one is financial risk assessment. So, we often hear statements like the like India's GDP is projected to be to grow at this rate in the current quarter and things like that. So, like basically we are trying to make some kind of a forecasting about certain economic parameters based on and on what basis are we making that forecast. Now, there are like the economies know certain predictors which are other in economic indicators based on which things like GDP depend.

So, they try to formulate this forecasting problem which can be either classification or regression like in the form of this linear problem using a linear function. And then if I may like use either linear regression or linear classification to actually come to make that prediction. And apart from this an important task is also variable selection. So, as I said like like in case of any forecasting task we may not know what are the actual explanatory variables or which variables are actually necessary for making that forecast.

So, we throw in a large number of variables. Now, if we use the lasso regression as I said it gives us a sparse solution that is most of the those predictors will have its weightage of 0. So, we can through those we so we understand that those predictors are irrelevant. So, we can ignore them and identify which are the relevant features. So, that not only helps in making better predictions, but it can also provide us with more economic insights. So, in conclusion supervised we discussed about the concepts of supervised learning and linear regression as well as classification.

We saw that it is both of them are based on a weight vector which can be estimated by minimizing a suitable loss function and we can specific structures like sparsity can also be imposed on it using regularization. linear classifier can be estimated using perceptrons or support vector machines which gives us max margin of linear that is it maximizes the margin between the different classes and hence it gives us a stable classifier. However, as we already discussed like the linear classifier will give us good results only if the data set is linearly separable which often it is not. So, we have to look for non-linear classifiers. So, in the coming lectures we will discuss the topic of non-linear classifiers which is which can be utilized using which can be realized using neural networks.

So, we will meet again in the next lecture. So, till then wish you like. So, all of them all of you please take care and we will meet again bye bye.