Transcriber Name: Saji Paul
Statistical Learning for Reliability Analysis
Prof. Monalisa Sarma
Subir Chowdhury School of Quality and Reliability
Indian Institute of Technology, Kharagpur

Lecture - 52
k-Nearest Neighbor Classification

Hi everyone, so today we are going to start a different classification strategy. Now, that we have already seen base classifier that was based on statistical matter. Now we will be seeing distance-based classification that is a k nearest neighbour classification.
(refer time: 00:41)

So, in this lecture we will cover the concept of k nearest neighbour classifications, how does this work? How does this kNN works and why we need to use kNN algorithm? How to choose the optimal value of k? What are the advantage? What are the disadvantage of the kNN and does kNN and have the curse of dimensionality problem this will be discussing these things in this lecture. Now the k nearest neighbour classified to start with.
(refer time: 01:05)

So, first of all, as I have already mentioned that we are discussing only supervised classifier under this in this course. So, we have already seen under supervised classifier there are different classification techniques, one is the statistical based standardized distance-based error based and distantary base. So, kNN is also a; we call it a k nearest neighbour classification technique.

It is a type of supervised learning classification and it is very widely used to solve many classification problem. The kNN is a data classification method for estimating the likelihood that a data point will become a member of one group to another, what does that mean? See here estimating the likelihood that a data point will become a member of one group or another.

Meaning, we try to find out the similarity index as I have discussed in my last to last lecture if I remember correctly. So, when we suppose there are two different classes of data. And we want to given a set of attributes we have to find out each bill this setup attribute means this data basically this observation belong to which class, class A or class B. What we try to this? We try to find out how does set of attributes this data is similar to which set of which class?

Basically, it is similar to class A or it is similar to class B. We try to find out it is similarity the index, less similarity. One way of finding a similarity is index is by finding out the distance Euclidean distance of these attribute values of this my test data point with the different classes the object belonging to the different classes. So, with the distance where the distance is smaller, so definitely my test data will belong to that class.

So, that is what we estimate the likelihood that a data point will become a member of one group or another it will try to estimate the likelihood. How we will try to estimate the likelihood? By finding out the Euclidean distance of this between these two object between these two observations. And based on this Euclidean distance we will tell this data point this stage data belongs to which group? Based on what group of data point is nearest to it?

We will classify the data based on the group which to which this data point is nearest and it is a distance-based classification technique that we have already mentioned.
(refer time: 03:46)

So, when I tell it is distance, how do I find out the Euclidian distance between two points suppose my attribute sets are the there are two attributes in a tuple in one observation suppose there are two attributes suppose this is. So, suppose a point that is x 1 y 1 and another point is x 2 y 2, if I want to find out the distance between these two points is nothing this all of us you know this how to find out Euclidean distance of two points.

So, if there is a two it is if the data is two dimensional. That means, if it has two attributes this is how we find out the distance that is $(x\ 2 - x\ 1)^2 + (y\ 2 - y\ 1)^2$ root of this whole thing. So, this now suppose if my observations or if my object has three attributes if it is a set of three attributes then we will have to consider three dimensions, so this is how we find out the distance.

We find that is $(x\ 2 - x\ 1)^2 + (y\ 2 - y\ 1)^2 + (z\ 2 - z\ 1)^2$ square root of the whole thing. So, similarly if there is a two dimensional, we will consider it as a straight line if there is a three dimensional will consider it a 3 dimensional plane and if it is more than three dimensional then it will consider it as a hyperspace. So, Euclidean in distance in n dimensional hyperspace if the n dimensional data points are this x 1 x 2 up to n y 1.

And the distance between them is same thing whether it is 2 dimensional, 3 dimensional, 4 dimensional, 5 dimensional just the same thing just you have to find out the distance.
(refer time: 05:23)

Now, how do you do it? Suppose, for a given set of data there are two classes suppose we given observation are there is just two classes. Suppose, here I have seen this is some this one class is this blue square which we see this is one class and another class this red colour triangle which is suppose this is another class suppose we have two classes. Using this training data, the training data or this observation data this data which is already we could observe.

You can I am using the term training if I will talk in terms of machine learning. I will use the term observe data if I will talk in terms of statistics. So, let me tell it is an observed data, so using this observed data we want to know the class of an unknown point. Now, my unknown point here is this green circle. So, greens I want to find out what is the class of this green circle how do I find out.

So, we have plotted this suppose the data are plotted suppose since it is each attribute has it is observation has two attributes. So, we can very well point plot it in a 2 dimensional way. So, now this if I plotted the data this suppose that plot takes this form. Now, what is the first step calculate, the distance of all training point from the test point what I will do is that. I will calculate the distance of all these training points how many training points are there?

Here, we have 3 4 5 6 4 8 9 10 11 total 11 trading points are there. So, I will calculate the distance of this green point from all this training point what is the distance of this and days distance of descent days distance of descent is this everything distance this this all the distance. I will calculate the distance of all the training points that is my then that is my first step. What is my second step? Find out k training points which are nearest to the test point.

Basically, once I have calculated the distance then I will arrange it in the ascending order. I have calculated distance then I will arrange it in the ascending order and then I will find out k training points which are nearest to the test point. I will come to what is k, let us k is some number then, I will find out k training point which are nearest to the test point, nearest to the test point means distance will be less.
(refer time: 07:52)

The k in kNN is a parameter that determines the number of nearest neighbours to include in the voting process. So, to find out as I told you here find out the k training points which are nearest to the test point. So, I will find out this k training point, now this k determines which k training points which are nearest to the data test data point. This k will determine how many of this number training data will be involved in the voting process.

Basically, this is in k nearest neighbour we just it is a sort of voting process. How many of the point is suppose there are? We will see the example here let is not talk now the test sample should be classified either to blue squares or to the red triangle yes, it has just two classes blue square and red triangle. So, this test is either has to be either should belong to here or to here if k = 3 we have found out the distance we have sorted in an order.

And now suppose I have considered k = 3. Suppose, if I consider k = 3 then what will k = 3 that is the nearest point the 3 that means the three nearest point to the green circle. So, this k = 3 means this portion and if for if k = 3 it is assigned to red triangle. If k = 3 what happens? There are how many direct triangles that is nearer to the green point nearer to that green point there are two red triangles and however nearer to the green point is only 1 blue square.

So, that means what when we if we take the voting process red will win because two will both for green and the only 1 blue will vote for green, so to green red will win. So, now that means the green circle belongs to the class which are these red triangles. Now, if k = 5 if I take k 5, 5 nearest neighbour. The 5 nearest neighbour means this dotted circle you see 5 nearest the neighbours in this case how many points are how many red points are closer to green.

There are 2 red points which are closer to green and how many blue points are closer to green,

there are 3 blue points that are closer to green. So, in voting who will win the blue points will win. So, that means the green point green circle belongs to the blue square cases. So, if k = 5, it is assigned to the blue squares, 3 square versus 2 triangle inside the outer circle. It does not consider the distance here if you see the distance red is very near the rates are very near it is not that.

We find out the nearest first nearest k and then out of this k how many the class which has more representation the test data will belong to that class. So, when we have considered k = 3, which class had more representation rate class at more representation that is it has 2 and blue class at 1 and when we considered k = 5 our blue had 3 representation and red has 2 representation. So, our test data will go to blue.
(refer time: 11:15)

So, if I now that is the step basically so, now if I write down the steps so first is load the data choose the k value. How to choose the k value? I will come. Now, for a test data point find the Euclidean distance of all training data samples. We have done that, store the distance on an ordered list and sort them in descending order sorry it is a missionary it is not descending order it should be ascending order.

We will store them in ascending order, so and then what happens; choose the top k entries from the sorted list. We will choose the top k address from this the sorted list top k means I should not say top k but the nearest the value which is less value. So, level the test point based on the majority of the classes present in the selected point. So, first what we will do we will find a equilateral distance of all the training points then we will sort the data.

How will you sort the distance? We will sort the distance and the ascending order and then we will consider the top k points, top k points means, the points which are the k points which are nearest to the; which has the smallest value basically and then among this k point which has the major representation our test data will go to that class.
(refer time: 12:47)

Now, this kNN is also called a lazy learner, why it is called a lazy learner? Because you see in all of the classifier or regression analysis whatever we have done regression and all those are also prediction classify also prediction all this is prediction only. Everything in all these cases you see we have developed a model, for all this we have once we get a training data then we develop the model.

So, based on that model given an input we try to find out what is the output? If it is a regression analysis given x what is the value of y that is what we try to find out if it is a classifier given x what is the class of this x that we try to find out. Whatever it is either it is regression analysis or classification whatever it is, so all clustering. So, now there we form a model first, but in case of this kNN we somehow do not form the model.

So, we whatever test data is given we just store the test data. So, from the statistical

perspective when the test data is there it is just there that is that we do not do any sort of calculation that is data is there. This data belongs to this class, this data belongs to this class. So, this data are there, so that is why this process is called lazy learning algorithm it does nothing when it does that when it gets the observation when it gets the initial observation is does nothing.

It only starts walking when it has to test a data when given an observation we have to find out the corresponding class then only it will start it is action before that it will not do any action. So, that means it does not learn any mathematical equation or model, it does not form any model, instead it just stores the training data. and starts computation when the test data is presented. So, that is why it is called lazy learning algorithm,
(refer time: 14:41)

Now, kNN is also a non-parametric method because, it does not make any assumption about the underlying data distribution. It is considered a non-parametric method because why not when we use consider parametric method, we have certain assumption about the underlying data this data has to belong to these distributions. Now, here in kNN is very much a non-parametric you do not have to be you do not have to bother about the distribution of the data.

And that is the input data that is the attributes of this different tuple. Simply, speaking kNN tries to determine what group a data point belongs to by looking at the data points around it. Given it the observation we have to find out it belongs to which class this class or this class I will just have to observe this training the different classes of data of the test data for test data whichever we find as test data is nearer to whichever class, we that our test data belongs to that class.
(refer time: 15:47)

So, now the condition equation is how do we find out the k value, I have already specified that we have sort the data then we take the first k value first k points to find out it is nearest neighbour for scale nearest neighbour. Now, this question is how to find out this k value there is not any specific way to determine the base k value. So, there are different empirically you can do find out take different k and find out the values.

So, on one way how is you can take it you might have to expand with a few values before deciding which one to go forward with. Means, you will try and deal with many values many k values and then you find okay this k value gives the better result, then we can go with this k. Now, how do we find out how will I experiment that how many k values will take and then which case giving good result how do we know.

Because, in the observations as does the supervised learning technique and the observation the class level is given. Now, and for the different observation for which we have to find out the class that is our test data basically it does not have the class level. So, if we are taking a whatever k value, we are taking 40 test data we are getting the correct value or we are not getting the correct value, we would not be able to say.

Because the states values data and class categories is not known to us. So, only way is that if we can find out the base k value if we have some chess data for which our class data is known to us. We have certain such data for which the class data is known to us and then we fix a particular k and then we try to for this data, we try to see it belongs to which category and whether it is correct because already we know the class level.

Now our exam this result is giving the result what we expected if the results matches then, this is good if the result does not matches then something is problem then that maybe we will have to change the k that is how we do it. So, that is what I have written here so, one way to do is by considering or pretending that the part of the draining sample is unknown. So, the observation what is given to us.

Or let us say if it is a training data or whatever it is doing that portion of it, we will consider as if it is not known to us, means we have not stored those data. We will take it a portion of this training data we will not store to a store we will just use a portion of the observation to make different classes. Now, this other portion which observation for which the class level is known to us, that we will try to put it into different categories taking different values of k.

Using this subset of training data states that are the true class levels are known in this state this data here the true class levels are known. Perform classification using kNN algorithm and for different values of k. Check the classification accuracies for different k will because, this class levels are known to us. This suppose we know this data x 1 x 2 x 3 it should fall in class A which is known to us.

Then again x 4 x 5 x 6 it should fall in class B it is known to us likewise we have many data we know which it is falling which class it is known to us. Now, we try to find out the in this x 1 x 2 x 3, we assume a particular k value and we see what which class we get. If we get class A then it is good whatever is whatever we expected we got that. So, likewise suppose we got we took 15 such suppose, we took 15 such data for which class level is known to us.

We took a particular k and accordingly we predicted the class then, we out of this 15, how many of us gave the correct result. We will see that, then again, we increase the value of k and we see out of this 15, how many of that gave the correct result? That way we will do different experimentation on k and will find now to choose that value of k which gives the highest accuracy this is one technique of taking the determining the best value of k.
(refer time: 20:08)

So, when dealing with a 2 class problem it is better to choose an odd value of k when there are 2 classes always it is advisable to choose an odd value for k. The value of k must not be multiple of the number of classes present these are some of the constraints. Choose the optimal value of k is by calculating root and where and denotes the number of samples in the training data set. One way of selecting k already have mentioned.

These are some other techniques you can one way is that definitely does the best method

which I have just mentioned. And another if you do not want to do so many because training data maybe you have around 220 training data which we want to use and try and find out the value of k. So, if many times you have to do it fine take one values of k, then okay you try it second value is, so that is definitely the best matter.

Otherwise, if you do not if you directly go for taking a value of k. So, these are the different steps at the; choose an odd value if it is 2 class A should not be a multiple of the number of classes. And k it is advisable to select k as a root of N where N denotes the number of samples in a training data set k with lower value such as k is 1 or 2 can be noisy and subject to the effect of outliers.

If we take a very less k value it can be noise and subject to the effect of outliers there are some variables like yesterday, I mean my last class I have given you an example suppose I have one class of cat another class of dog. So, cat also has but to say 4 legs 1 tail dog also have 4 legs 1 tail. And some dogs again there is some dogs whose height is almost same as the cat but usually dog's height is more than a cat.

So, this is not an out layer but there are some dogs with height is less but I am just to give you an example. So, likewise there will be some sometimes some attributes which basically we get some value of the attributes because of some out layers which is usually not normal because of some abnormal situation or maybe because of some very rare situation we get those value. And if we select k value very small suppose, 1 or 2 we tend to we may select a round classes because of this outlier values.

We will find a very good match with the first data because of this out layer attributes maybe actually that is not the class of that. So, it is always advisable to take a higher value of k. On the other hand, came with larger values in most cases will give rise to smoother decent boundary but it should not be too large as well if you take k value too large the whole sample size generally the whole training data sets.

Then, what will happen lots of calculation which will take a long because; calculating Euclidean distance for each and every point it will be very computationally intensive. So, you should not be very large also but then a large value of k is advisable it gives a very smoother boundary decision boundary. Otherwise, groups with fewer number of data points will always be out voted by other groups.
(refer time: 23:28)

So, some pros and cons of kNN pros it is easy to understand and simple to implement we have seen how easy to understand it just find out the training data sort it find out the k nearest number, which is the how many representation the class which has the highest representation test belongs to that class. It is so easy implementation. It is ideal for non-linear data since there is no assumption of down drilling data, data can be any format.

Since, there is no underlying assumption these classifications that work very well for non-linear

data as well. It can naturally handle multi-class cases; it can perform well with enough representative data. Then, what is cons associate com associated computation cost is high as it stores all the training data it has to store all the training data it has to find out a Euclidean distance of all the training data sources computation cost is very high.

Requires high memory storage need to determine the optimal value of k. Prediction is slow if the value of N is very high and it is sensitive to irrelevant features. Sometimes some classes have some irrelevant features, some very uncommon features and it becomes what a bad thing is that it becomes very sensitive to those irrelevant features also. We consider those irrelevant pieces also important.
(refer time: 24:53)

And the term curse of dimensionality means, if the data is of high dimension. So, if the data is of high demands high dimensional data high dimensional data means many attributes for it a tuple has many attributes suppose a tuple has 100 attributes then what happens? We have to find out the distance you can imagine the time it will take to find out the distance of these 100 attributes.

So, this kNN and suppose from the curse of dimensionality. With the increasing number of features time complexity of the distance computation largely increases. Obvious, does not it we have to find out a distance of such high dimensionality our disc competitor it is very computationally expensive. kNN does not work well if there are too many features. Hence, diamond is dimensional reduction technique like principal component analysis, feature selection must be performed during the pre-processing.

So, if you are interested in using kNN classification algorithms if the number of features are used. Usually, that is the case usually it is same, there are a number of features a huge then what will happen we will have to reduce the features there are different methods of reducing differences. We will have to reduce the features during data pre-processing, principal component analysis which I will not be discussing in this lecture.

So, way of written feature reduction is that one way is principal component analysis basically from the different suppose there are 100 features. Out of 100 features there is some data some features which are redundant. Again, there are; some dependency in some feature set. So, suppose let me tell you suppose there are feature set suppose say X Y Z A B suppose these are the different features and for one observation we have total 5 features for one observation.

In this feature suppose if we have seen if X increases Z also increases or if X increases Z decreases. So, there is a dependency between these 2 features, when there is a dependency between these two features. Then, why we will consider these 2 features, that means this feature will have an effect on the class that we understood. This feature means to find out the class to find out the class which class it belongs to we will need to consider this attribute.

We have seen we will need to consider this attribute but if we consider only X that is sufficient because, X and Z are very much correlated either positively correlated or negatively correlated.

So, we can just ignore one though this we can find out given the features that we can find out which of the attributes are correlated. If the features it attributes are correlated, we can ignore taking all the attributes that is one way of reducing the features.

So, that is how basically if there are too many features again there are some features which have which does not have any effect on the class level. So, why unnecessarily drag that features unnecessarily find the distance these features. So, there are different ways how do we find out how do we reduce the different features different attributes it. So, once we reduce the attribution then kNN becomes quite good. Then, it does not it is main problem is the cost of dimensionality.
(refer time: 28:16)

So, now coming to the end of this kNN classification we learned about a distance based classifier which we call it a k nearest neighbour method. This is a very simplest classification technique in fact from all the classification this is the simplest one. In fact, for base theorem also I will not say it is simple as we will have to calculate different probabilities. So, and it is widely used in many applications.

So, we learn the kNN classification method the salient properties of kNN it is advantage and it is disadvantage. So, now is the time for tutorial in the next class we will have a tutorial before going to other classification strategies.
(refer time: 28:55)

So, these are the references, thank you guys.