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 - 57
SVM (Part - Ⅳ)

(refer time: 00:30)

Hi guys, so, in continuation our lecture on SVM today what we will discuss? We have discussed about linear SVM binary classification, is not it? We have seen only binary classifications on for two classes we have seen linear SVMs linear means where data are can be linearly separable. In that also we have seen for case of multi-class also in multi-class if the data is a linearly separable then what we can do how we can use linear SVM that also we have seen there are two strategies OVO and OVA strategy.

We have seen and we have seen the pros and cons of motor strategies. Today we will be thing for non-linear data, non-linear data what should we do? We should use linear SVM or non-linear SVM or if you use linear SVM then it is the same way will be used or will be changing something that will be discussing. Basically, we will and one more thing which we will discuss is that when for non-linear data now we try to make it linear.

So, for that we do a transformation that is called phi transformation that also will be discussing in this lecture.
(refer time: 01:39)

So, talking of non-linear support vector machine basically support vector machine is classification for linearly non separable data, what how we do? For linearly non-separable data so what sort of support vector machine do we use. So, here you can see in the figure there is a this so now this is here you see this this figure this is a linearly separable data. So, there are two class plus class and minus class which we can easily separate by using a hyper plane.

Now you see in this figure B you see the two data's there we can we cannot use a hyperplane to classify this to data because they are not linearly separable. So, in general if data are linearly separable then there is a hyperplane else if it is not linearly separable, we then we basically call it a hyperspace.
(refer time: 02:39)

So, such a linearly not separable data can be classified using two approaches. So, now we want to classify data which cannot be linearly separable. So, how we can do there are two approaches one is using linear SVM only but somehow twisting it twisting a bit in this or the other way. So, we will be using linear SVM only and another is we will use non-linear SVM that is now we how can how we classify data, we classify data using a hyper plane.

So, if in non-linear hyperplane we will not be using a hyper plane so we will see how we do that. And next the most comfortable technique for us will be because we are compatible with linear SVM if somehow, we can use linear SVM for separating this non-linear data.
(refer time: 03:29)

So, first we will see that then we will go to non-linear SVM. So, here in the example you see if these two sets of data you see. This is one class this is the other class so this is very much linearly separable bearing few data few data. So, here just we have two data these two data's are somehow if we use the hyper plane test data we cannot separate classify it into the two categories. So, this will be basically we can say classification error.

So, in this sort of if this is one point and this is the other point if the points are in this way, then basically, we cannot say that there is a linearly separable data. But there are very few cases in this example there are some such types where the very few data's which are which cannot be linearly separable. In such cases what we can do is that we can do the classification giving ignoring this data which cannot be ignoring just few data's which cannot be classified.

So, this is basically we are accepting some classification error because of this data and this data it cannot be if you use a suppose we have used this hyperplane red colour line if you see. If you have used this hyperplane along with this boundary this is one boundary and this is another boundary. If we use this boundary then what happens? So, this data will be misclassified. So, accepting misclassification for some data we have we have gone with the linear SVM.

So, linear SVM means we have found out that many maximum margin hyperplane. Now we will come back to this. Basically, these misclassifications how much misclassification we can afford that we will specify for that there is a parameter how we can specify, how much misclassification we can afford. So, essentially what I am going to say is that when we try to construct a SVM for linearly separable data we what we do is that we have a minimization function and subject to some constraint.

So, what was the minimization function our minimization function is magnitude of w / 2 that was

the minimization function. Now along with this minimization function what we will do? We will add another one component, another one constraint that is basically the penalty factor for the data's which cannot be classified if you use a hyperplane. So, when we use this penalty factor what happens? Our hyperplane becomes narrower and narrower.

Our margin becomes narrower and margin may be very less. It depends on how much classification error we can bear as well as how much we can what is reduce the MMH. So, it is basically a trader between the MMH and the classification error. So, we introduce another one factor that is we call it the penalty factor and this sort of changing the margin we call it a soft margin. The previous one what we do in the general case that is called hard margin.

And now when we change it this is called a soft margin. So, this I will not go into the details of soft margin because soft margin method is not very much applicable. Because when we see where some data's are not linearly separable. So, these are not a few data's and usually there are too many such data. So, when there are too many such data this soft margin does not work well, sub margin would have worked well if there are only few number of data which are not linearly separable.

Then we can reduce the margin by using the penalty factor and then we can classify data with good results. But since in reality that is not the case so soft margin is not it has not much applications. So, we will not go to the details of soft margin that is what this can be done by reducing the margin, soft margin is this concept. So, we will not go to the details of that.
(refer time: 07:14)

Now we will try to add how to say classify the non-separable data by some other means we will see that so, what linear SVM undoubtedly β to classify data if it is trained by linearly separable data obviously. So, if we have developed an SVM if you have developed a machine by using our training data is linearly separable data. For developing the machine, we have our we have used training data is not it.

Basically, if I say in other terms basically when I have computed the function that is the SVM that is y = f x this function when I have computed a function, I have used some observation. Now if these observations are linearly separable data, then in that case linear SVM is better to classify, it is better to classify any sort of training data which can be what to say given any training data. Now we can use this SVM to classify it efficiently.

Now linear as SVM we can also be used for non-linearly separable data provided that number of such instances very less that we have seen by using soft margin provided that such instance is very less. This is important. Linear SVM can also be used for non-linearly separable data

provided at number of such instances less meaning that in that case we can go for soft margin by using a penalty factor basically by trade-off between the margin and the misclassification error.

However, in real life application number of data overlapping is so high that the approach cannot cope to accurate classifier. So, soft margin does not give us an accurate classifier. As an alternative to this there is a need to compute a decision boundary which is not linear. So, if that is not possible. So, soft team soft margin concept if that is not possible that we do not get a good classifier our classification visual is poor then so to compute a decent boundary.

So, we need a non-linear distance boundary in that case because this one a linear decision boundary only can classify between linearly separable data. If the data are not linearly separable then we will need a non-linear boundary. So, non-linear boundary we also call it a hyper surface.
(refer time: 09:37)

So, here is an example this is a linear hyperplane this you can say this is a high non-linear hyperspace, this is a non-linear hyperspace. So, note that a linear hyperplane is expressed as a linear equation in terms of n dimensional component, is not it? A linear hyperplane we use we express in times in terms of a linear equation w x 1 + w x 2 + w x 3 + b = 0. So, it is very much a linear expression, is not it?

A linear equation in terms of the n dimensional component. How what is the attributes of the each data point? So, if the if a data point is two dimensional so it will be a linear equation and two dimensional and if it is a three dimensional then will be having the linear equation in a three dimensional that way. So, now if it is a non-linear hyperspace then it will be our equation will also be non-linear.
(refer time: 10:32)

So, we can express a hyperplane in this way we have already seen that. Whereas a non-linear hyperplane may be a non-linear hyperplane we can express in this way maybe. So, w 1 x 1 ^{2} w 2 x 2 ^{2} so it is basic suppose it has a dimensional is three, x 1, x 2, x 3 x is a three dimensional data. So, that means each observation has three attribute x 1, x 2, x 3. So, and if it is non-linear suppose the non-linear expression is this in this form nonlinear it can be any form.

So, w 1 x 1 ^{2}, w 2 x 2 ^{2} plus this is anything it can mean any form non-linear basically. So, suppose let us consider this is a non-linear hyperspace. Now what is your task now? It is to find a non-linear decision boundary. So, we have to find a non-linear decent boundary so non-linear descent boundary means we need to find a non-linear hyperspace. You know that is a non-linear hyperspace in input space comprising with linearly non separable data.

So, you see here this is this data's are not linearly separable so we need a hyperspace. So, as to separate this linearly not separable data. So, now these tasks the stocks of using if the data's are non-linearly separable now the as I told you if the data is a nonlinear separable we cannot use a linear hyperplane. Now that means we need to find a non-linear hyperspace. Now with how to solve this for this non-linear data?

This is something not very hard and also not very complex because there are we can basically use the technique whatever we have used for linear SVM we can extend that technique. Of course, there is one way of using non-linear that is different way but another way is that we can also extend that method what we have learned for linear SVM we can also extend that for linearly not separable data. I am not talking about the soft margin.

Soft margin also basically we use the linear separator only, linear hyperplane only, linear hyperplane but not that. Even for data for which is not linearly separable we can extend a linear SVM to solve to classify that this data. So, let us see how we will do that.
(refer time: 12:55)

So, this can be achieved in two major steps, what it is? Transform the original non-linear input data into higher dimensional space as a linear representation of data. So, our original data was non-linear here we see this data, this data is non-linear. So, this data is a non-linear in how what is the dimension? It was in a three dimensional space. So, transform this non-linear three dimensional space into a higher dimensional which how we will transform so that it becomes linear.

This is non-linear so with the we will do the transformation in such a way of course this transformation will be in a higher dimensional this linear we will transform into non-linear sorry it is non-linear will transform it to linear. So, note that this is feasible because SVM performance is decided by the number of support vectors not by the dimension of the data. Already we have seen in my last lecture that the performance of the SVM that is the complexity of the SVM.

I have already mentioned complexity of the SVM is dependent on the number of support vectors. Remember the summation of λ i y i x i is there. So, it is complexity is totally dependent on the number of support vectors, it is not dependent on the dimension of the data. So, if the complexity would have been dependent on a dimension of the data so if we increase the dimension definitely so our complexity would have increased enormously.

So, that is then that would not have been a feasible solution. Now since SVM performance is

not decided by the dimensionality of the data it is decided by the number of support vectors. So, if we increase the dimension also it will not have a very higher impact. Of course, they are definitely high dimension it goes to higher different dimensions definitely our calculation will be higher but then it will not make an impact on the complexity of the SVM.

Search for the linear distance boundaries to separate the transform higher dimensional data. So, first we will have to transform it the non-linear data into a linear data and then once we transform the nonlinear data to linear data then we will have to search for the linear distance boundaries which will separate the transformed higher trans dimensional data. Non-linear data we will convert it to linear data in a higher dimension.

So, now once we have done that, then what we will have to? We will have to search the decision boundary that is the hyper plane that will separate this transformed higher dimensional data. This we can do in the same way how we have done for linear SVM.
(refer time: 15:36)

In nutshell, to have a non-linear SVM the trick is to transform non-linear data into higher dimension than linear data. This transformation is popularly called non-linear mapping or attribute transformation or phi transformation. This transformation it is called non-linear mapping or attribute transformation or phi transformation more popularly it is called phi transformation.

And then once we transform it rest is same as the linear SVM what we have learned. So, now let us consider a second order polynomial. Say this is the let us consider second order polynomial in a three dimensional input space. So, this is the second order, is not it? So, we have a quadratic term so it is a second order we have a quadratic term. We have this x 1 x 2, x 1 x 3, x 1^2 so it is very much a non-linear polynomial.

So, it is in the what is the input space? Input space is three dimensional so we have x1, x 2, x 3. Now this thing we will convert it to linear in a higher dimension. How will convert it? See here.
(refer time: 16:42)

The 3D input vector this is a vector can be mapped into a 6D space that is Z using the following mappings. This 3D we have mapped with six dimensional space, how? z of one is phi of x that is x 1, so z of 1 is x 1, z of 2 is x 2, z of 3 is x 3, z of 4 is $x\ 1^2$, z of 5 is x 1 x 2, z of 6 is x 1, x 3 all the terms that we have all the monomials that we have here. What are the different monomials we have that mono means basically we have used another one what to say attribute and that is this attributes in a higher dimension.

See here total we have six monomials so we have used six different attributes and so this is

how we have done the transformation. So, now our data X the vector x 1, x 2, x 3 it has become X has become Z where attribute of X was x 1, x 2, x 3 now attribute of Z has become z 1 to z 6. (refer time: 17:51)

So, now basically now the transformed form of the linear data so now this is my transform formula with linear data. I have just replaced the different x 1 and x 2 different form instance of X with Z. Now this is a very much a linear, is not it? Very much a linear equation. Now this is we have to find what is it, we have to find is MMH for this hyperplane, MMH means basically what we need to find out?

We need to find out a different W value we need to find out the Z value and that is all and B value. So, thus if that Z space has input data for its expression x 1, x 2, x 3 and hence Z values then we can classify them using linear distance boundaries. So, this equation we have same method we will be using Lagrangian multiplier method. For Lagrangian multiplier method now what will be our optimization function? What will be our but to say constraint?

So, accordingly we will be what to say we will solve that using the constraint and the Lagrange and multiplier that is del L / z of I del L / B and accordingly we will get the values of z, B, w. Once we get all the values of z, B, w then from this we can find out the we found out all the values of z, B, w then what happens then basically we found out the SVM. Now given any X so that X is in a 3D space that X will convert it into 6D space.

Now once we converted it into six dimensional space so what is x because for this we have we found out the SVM means we found out all the parameters below. So, we then we found out the SVM so once we found out the SVM then once given X will be able to find out which belongs to which class, same. What linear SVM we have done? Same technique we have done. Now just that now our expression has become this that is all nothing got changed.
(refer time: 19:55)

So, just is an example. See here in the figure is so an example of a two dimensional data set consisting of a class level plus and minus that is a not linearly separable as you can see very nicely. So, this is these are plus class, this is minus class definitely it is not linearly separable. (refer time: 20:19)

Suppose these data's are given and somehow we have found out the function, we have found out the w 1, w 2 value, b value and everything. So, suppose this is the equation, we see that all instead of class minus can be separate from instead of class plus class by circle, is not it? We can separate it by a circle which is a hyperspace. So, without anonymity say the following equation of the decision boundary can be thought of.

So, this is the equation of the distant boundary. We have done the we have done all sort of what to say transformation and like Lagrange multiplier and then we will find out the W value, b value everything and then we will find out the decision boundaries. So, now here since this is circle so we can think that maybe this is maybe the equation of a circle. So, now what happened?
(refer time: 21:17)

So, this is my distance boundary, is not it? This is my hyperplane. So, any data which is in this hyperplane will be this will be equals this minus will be equal to zero, any data which is above the hyperplane will be get a positive value any data below the hyper plane will get a negative value. So, this is the expression. Now this is very much a non-linear representation, is not it? So, now this non-linear representation so we need given a test data, we have to find out.

Because we need to find out all this value, is not it. So, given a test that are done we need to find out whether it falls in which data so for that we need to find out a hyperplane. Now for finding out the hyperspace since this is a non-linear equation separating this two. So, this non-linear equation basically we can find out the hyperspace only. So, if you are interested in finding on the hyper plane so we will be using the linear SVM.
So, what we will do we will just transform this data. So, how we have used this transformation? Suppose for this we have used $x\ 1^{2}$ - x 1 we have user transformation that is z 1 and for this $x\ 2^{2}$ - x 2 this 2 term we are talking it together and we are taking another one the transformation that is z 2.
(refer time: 22:38)

So, here there was it was a two dimensional data, here I have we have transformed into two dimensional only. It is basically the transformation how you do the transformation. Here without any ambiguity, we can do this transformation, is not it? Because phi x just z 1 we can write it in this pattern because $x\ 1^{2}$ - x 1 there is no other separate coefficient also. So, very well I can write it and this also very well I can write it phi to x as that is a z 2.

So, now what is my this is my z 1 + z 2 = 3.5, so this is my hyperplane. So, once I know my hyperplane then I will happen I can easily given any data I will be able to tell it falls in which class. So, it was in non-linear X is in 2D space which is non-linear now we have convert Z this is also into this space and now this has become linear.
(refer time: 23:35)

The non-linear to linear transformation however there are some issues. When we have transformed this long linear to linear transformation however there are some issues like recent this last program of problem only you have seen. Here see I have used $x\ 1^{2}$ - x 1 is equals to that is one transformation I have used. Why did not I use one transformation for $x\ 1^{2}$ and for x 1 another transformation then again for $x\ 2^{2}$ another x transformation, x 2 another transformation.

So, basically why did I use that now cannot I use that. So, these are some simple questions like. The non-linear mapping and hence a linear descent boundary concept looks pretty simple but there are many potential problems. Whatever how to choose the non-linear mapping to a higher dimensional space that is our how to choose? For the last example the way I have chosen is that better or if I would have chosen instead of z 1, z 2 if I would have chosen z 1, z 2, z 3, z 4 that would have been better which should have been better?

So, for that we will have to do many again experimentation and find out. So, how to choose the non-linear mapping? In fact, as the phi transformation works fine for small examples but it fails for realistically size problems. So, if we do the phi transformation and if there are n dimensional this n dimensional only input space is n dimensional and n dimensional is only quite say high. So, then converting this n dimensional to higher dimensional it may be quite higher.

So, higher you can see here. For n dimension input instance there exists this many number of different monomials. What a monomials? If it has this two dimensional x 1 and x 2 different monomials can be x 1, x 2, x 1 x 2, x 1^2 x 2, x 2^2 x 1. So, different types what I am selling is suppose I have x 1 and x 2. So, different monomials can be x 1, x 2, x 1 x 2, x 1^2 x 2, x 2^2 x 1 then x 1^2 x 2^2.

There can be different monomials so for each monomial if I use different transformation accordingly my dimensional increases. So, the total number of this is the total number of monomials we will get for a dimensional of D sorry for a maximum degree of the monomial sorry for the dimension of N and if D is the maximum degree so this many monomials we get. So, for this many monomials if we try mapping this each monomial if you try mapping to a different attribute.

So, accordingly our dimensionality increases. So, if a dimensionality increases so what happens? Though SVM as I mentioned it does not support from curves of dimensionality but still one of the dimensionality becomes very high. So, the computation increases computation the number of computation increases when we do the dot product. So, number of additional number of multiplication and number of additions increases when I do w i W dot x.

So, the computer computational it becomes computational intensive. So, cost of mapping this is one of the issue.
(refer time: 26:54)

So, that is the dimensionality problem. It may suffer from the curse of dimension often associated with high dimensional data which I have just already mentioned. More specifically in the calculation of W dot X or X i dot X when we try to find out the class of X then we do X i dot X

is not it, the dot product. So, if it is of higher dimensional what happens? When you do X i dot X how many multiplication and addition we do?

If it is an n dimensional data for n dimensional data, we do n multiplication and add n addition. If accordingly, if our dimensionality increases the number of multiplication number of additions increases. As the number of input instance as well as the support vectors are enormously large it will be computationally intensive. So, that way it will also suffer from the dimensionality problem.

So, again solving the inequality constraint optimization problem in the high dimensional feature space is very computational intensive task. Now when we have to find out the values of this different parameters using the Lagrangian multiplier method what we have seen. So, when using this inequality constraint optimization problem when we solve it for high dimensions space it becomes very computationally intensive task.

Because they will be getting many expressions many equations, is not it? If the n dimensional so we will get when we do del L / del x i. So, if this n dimensional we get n expression. And similarly, if there are more than if the dimensionality is more the number of equation is more. If the number of equation is more than accordingly solving this equation will also be very computationally intensive.
(refer time: 28:31)

Fortunately, mathematicians have cleverly proposed an elegant solution to their work problem. Now discursive dimensionality it would have been a really a case to SVM because non-linearly separable data if we can use linear assume that is very good. Why unnecessarily complicate the issue? Where linear SVM is very simple. So, if you can use it in linear SVM then it is definitely very good but linear SVM we have seen two technique.

One is soft margin which do not which do not work well. If the number of data which are basically which are very near to the hyperplanes number of such data is large so it does not work well, we have seen already and moreover. So, other technique was data transformation. If data transformation if for a while doing data transformation if we face the case of dimensionality then the advantage of data transformation is gone.

Then no point using the linear SVM then we should basically go for nonlinear SVM only. But mathematician has cleverly proposed an elegant solution. So, this cause of the dimensionality problem we will not be we will this even if we do data transformation so we will not face this dimensionality issue. The solution is like dual formation of the optimization problem and kernel trick. So, what is the solution?

The solution as it is and there is a concept called dual formation of the optimization problem whatever optimization problem we had. What is the optimization problem? Minimize W / 2 to a constraint, is not it? So, that was our optimization problem. So, we will find a what to say dual formulation of that which is much easier to calculate. Because here we have seen in this optimization problem when we do del L / del x i and we will have to find out the del L / w.

So, all the number of equations increases tremendously. So, if we can find out some sort of formulations which will reduce the computational thing task in the while what to say while finding this inequality constraint optimization while you are trying to find out the maximum merge in hyperplane while trying to optimize this if you can reduce the computation so then it will be much beneficial for to use the linear SVM for non-linear SVM data.

So, that is called dual formation we will see that in our next lecture. Then and there is another concept called kernel trick that also helps in this what to say becoming this transformation. It has as we have seen it would have it faces the case of dimensional problem but how does kernel trick and dual formulation of automation problem will help in what to say improving the performance of this, we will see that in the next lecture.
(refer time: 31:33)

So, in this lecture we have introduced the concept of linearly separable data and linearly not separable data. Then we have learned about linear SVM, how we can use linear SVM to classify non-linear data. We have seen two methods but soft margin we did not discuss of course. Further we learned about the idea of non-linear to linear mapping to handle non-linear data that is the phi transformation.

So, in the next lecture we will be discussing this kernel trick as well as this one dual formation of optimization problem. This will be discussing in our next lecture.
(refer time: 32:11)

So, thank you guys.