NPTEL

NPTEL ONLINE CERTIFICATION COURSE

Introduction to Machine Learning

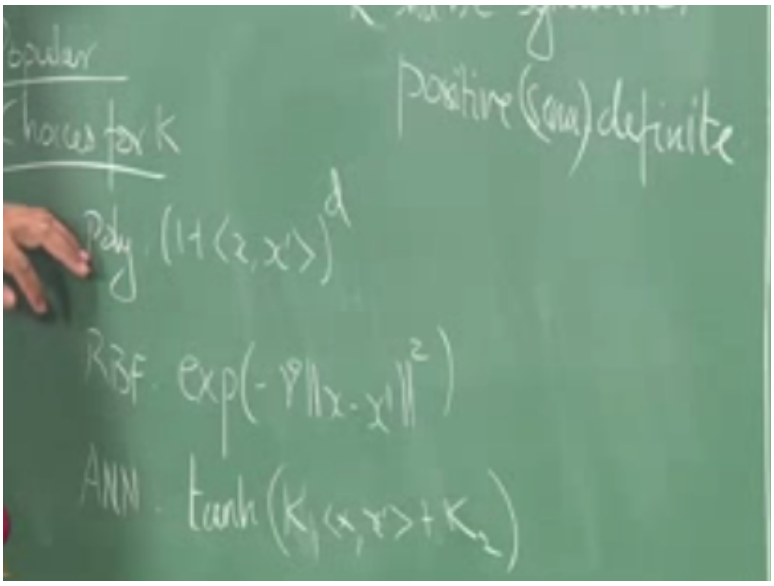
Lecture 30

Prof. Balaraman Ravibdran
Computer Science and Engineering
Indian Institute of Technology Madras

SVM Kernels

So if you remember I asked you to note the fact that I am using a inner product they are right xi transpose X is less the inner product of two vectors and the way I wrote the dual also I had only inner products in there right so in fact if I want to evaluate the duel I need to only know the inner products of the two vectors likewise if I want to finally evaluate them use the classifier that I learn I still need to only find inner products right.

(Refer Slide Time: 01:24)



So if I can come up with a way of efficiently computing this inner products right I can do something interesting so what is that so what do we normally do to make linear classifiers more powerful basis transformations this is somebody said basis transformation right so I can just take

my hedge right replace it with some function H of X that gives me a larger basis right it could be just replace it with the square right.

I take X and replace it with x^2 right and then I will get a larger basis and now it turns out that i can do a lot of math but I can get up get with get a duel that looks like this so that is the inner product notation right so if i can compute the inner product so i can just solve the same kind of optimization problem right but I can do this in some other transformed space okay.

So likewise our f of X is going to be so essentially what I need to know is Hof X for whatever pair X and X prime that i would like to consider right so in the training it is the paths of training points right while I am actually using it is one of the support point and the input data that I am looking at right so at any point I just take this pairs of data points and I need to compute the inner product right.

So I am going to call this as some function way which is a kind of a distance function or a similarity measure between H of X and H of X prime right such similarity measures are also called as kernels right so kernels or nothing but I mean so we might have heard of kernels in the context of support vector machines we have been trying to use this VM or any of the other tools for some projects over the summer you have heard of kernels right kernels are nothing.

But similarity functions okay so the nice thing about the kernels that we use right is that they actually operate on X and X prime okay they operate on X and X prime but they are computing the inner product of H of X + H of X prime did you see that they are going to work with X and X prime right but they will be computing the inner product of h of x + H of X prime.

So I will give you an example so the kernel function case should be symmetric right and positive semi-definite okay positive definite semi definite is fine in some cases positive definite people number remember what positive definite is right, right and that if it is semi definite it essentially we want the quadratic forms to be to be positive.

We do not want to take X transpose ax and suddenly find it is negative so it is in fact you remember I told The X transpose AX is usually the quadratic form that we are trying right and that will actually messes up big time in the computation if the quadratic form becomes negative then we love problems in all the optimization thing going through okay so that is the, the mechanistic exam reason for wanting it to be positive semi-definite.

There is a much more fundamental reason for it which I have not developed the math or the intuition for you to understand so it has to come at a later course so hopefully in the kernel methods course if you are taking it you will figure out why that is needed okay so there are many choices which you can use for the kernels so the something called the polynomial kernel which is essentially one plus inner product of X, X prime the whole raise to the power D so D is a parameter you can have D of two three four you can even have D of one is essentially here whatever we have solved.

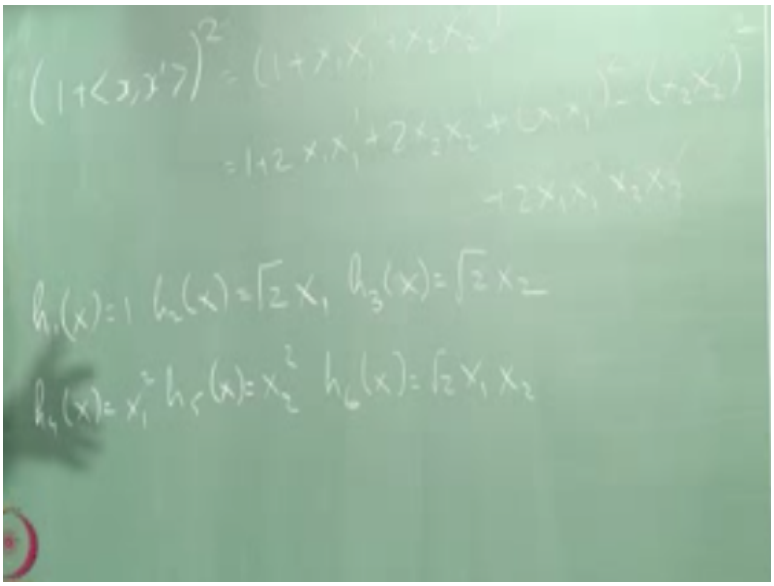
So far right you could have D you two three four whatever and this h_1 is some called the Gaussian kernel or the RBF kernel right so where the, the distance is given by $E^{-\gamma}$ X-X prime square norm of X minus X prime squared is essentially the Gaussian without here the normalizing factor right so that is why it is called the RBF kernel so if you want to call it the Gaussian kernel you actually have to make it Gaussian otherwise called the RBF kernel.

And then this is called the, the neural network kernel or the sigmoid kernel sometimes not just the hyperbolic tangent right this is coppa one and Coppa two some constant some arbitrary constants k1 and k2 which are your parameters attitude and this is X, X prime inner product okay so these are some of the popular kernels which can be used for any generic data but then depending on the kind of data that you are looking at right where the data comes from people do develop speech the specialized kernels they for examples for string data people have come up with a lot of kernels.

When you want to compare strings how do I look at similarity between strings so the nice thing about whatever we have done so far is that you can apply this not just to data that comes from RP right you been assuming so far that your x comes from some dimensional real space as long as you can define a proper colonel right you can apply this, this max margin classification.

That we have done to any kind of data does not have to come from a real-valued space okay which is not true of many of the other things you are looked at right all those inherently depend on the fact that the data is real value right because of this nice what is called the kernel trick right so you could do all of this nice things so as long as you can define appropriate kernel that you can actually apply this to any kind of data so that is one very powerful idea.

(Refer Slide Time: 09:28)

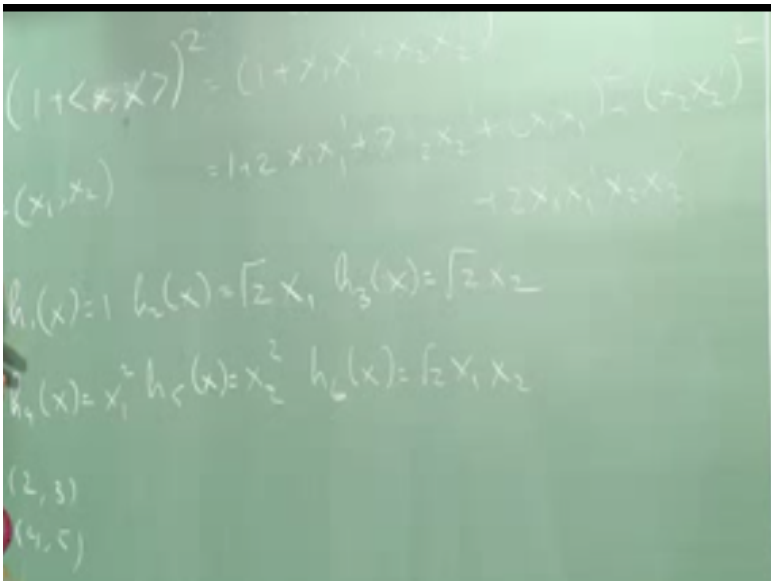


So just to convince you so let us look at the polynomial kernel of degree two right operating on vectors of two dimension okay there are 2 2's here ok so the degree is two the D is two and the P is also two but they need not necessarily be the same that I could have had a much larger thing over to CC for me to write something so this is what 1+ right now just squared it now if you think of h plus the following right.

So what is this function H it is essentially the quadratic basis expansion right so i have two, two features x_1 x_2 right so i give so remember that X, X is X_1 X_2 right they consider of two coordinates x_1 and x_2 right so this is essentially the quadratic expansion the first thing is one the second coordinate is x_1 third coordinate is x_2 so it keeps it as it is okay then fourth coordinate is x_1^2 first coordinate is x^2 the sixth coordinate is x_1 x_2 it is done all the quadratic basis expansion right.

Now if I make this operate on X and X Prime and take the inner product so what will be the terms 1,2 x_1, x_1 Prime 2 x_2, x_2 prime X_1^2 x_2 x_1 prime squared right x_2, x_2 prime the whole square + 2 X_1 X_1 prime X_2, X_2 prime is exactly what we have here right so what is the nice thing about it is I can essentially compute the inner product of X 0 X and X prime first add 1 and square it so numerically what I will end up with is the same as what i would have ended up with if i had done the basis expansion right and then taken the inner product that makes sense right.

(Refer Slide Time: 13:05)



If i had just taken whatever is a vector so original vectors let us say I have some 2, 3and 4, 5 so instead of doing this basis expansion and then computing the inner product right I can just take the inner product right away right this license t_2^2 this answer so this well for degree to it might not seem great what about degree15 polynomial I have essentially doing similar amounts of computation except that I have to rise something to the power of 15 right so is any questions.

So far any questions so far they are all good there is expansion Air services expansion is if you if you thought something else about basis expansion please correct it this is business expansion right so i take the original data okay I then since I said you could have a new component set or sine x cos x_i mean does not matter right you could think of variety of different ways of expanding the bases in this case I am just doing the quadratic basis expansion right is it clear to people.

So whatever we have done so far and so this whole idea for kernel and other things are arriving rather straightforward so what I cannot right now for you is what is the basis expansion for the RBF kernel it turns out that the computation is doing is actually in an infinite dimensional vector space okay so here the computation is a six dimensional space and I took some data point from a two dimensional space computation in a six dimensional space right.

And I gave you back the answer but all the time doing computation only in a two-dimensional space and I only took the inner product of these two and then added 1^2 so I am essentially doing computations only are too right well the actual number i am returning to you is the result of

computation done in our six that is why it is called the kernel trick right so likewise the RBF kernel I will do something in whatever is the original dimensional space.

You give me but the resulting computation has an interpretation in some infinite dimensional vector space case it is not even easy to write it down so that is why the RBF kernel powerful they work on a variety of data right but they are not all powerful this have to be careful about it right so, so that is all there is to support vector machines so we have done this support vector machines as well.

So I don't know if people who have used lib base VM or one such tool for that for most RBF kernels you would have to tune two parameters one is C which we already saw right that is essentially how much penalty you are giving to the thing other one you will tune is γ essentially this right it is some kind of a width parameter for your Gaussian this how wide you are Gaussian is it just it is control set so that is γ so those are the two parameters you tune and for polynomial kernels you have a D and you have your C right and for Sigma L kernels you have constants k1 k2 and you have C and this form of defining a support vector machine is called as CSVM okay.

There are other ways other constraints that you can impose on it not just the penalty on the Z's you can impose penalty on the number of support vectors you consider right you want to say so suppose I run the data and it comes back and says okay everything is a support vector right so that is not something interesting how can everything be a support vector can all the data points be equal distance from the separating capita not if you are considering linear but when I am considering RBF kernels right the separating hyper plane can be very ,very complex right.

So in which case you might end up with a lot of support vectors typically i do not know if you have not thought too much about it and you are setting some very high values for C and trying to run this thing you will end up with like sixty percent of your data as being support vectors so instead of trying to do that empirically second on why only 120 support vector so let me try different see differential γ and so on so forth you can use something called the new SVM new not new the Greek new SVM which gives you a upper bound on the number of support vectors you are going to get you can say do the best you can but do not give me more than 30 support vectors something like that to that effect okay.

Funded by
Department of Higher Education
Ministry of Resource Development
Government of India

www.nptel.ac.in