NPTEL

NPTEL ONLINE CERIFICATION COURSE

Introduction to Machine Learning

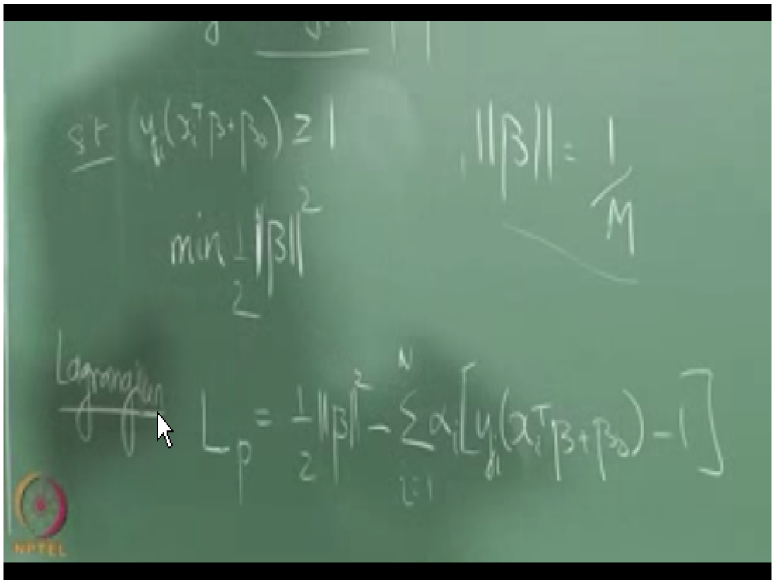
Lecture 28

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

Support Vector Machines II
Interpretation & Analysis

So this is the optimization problem which is actually a simple optimization problem is a quadratic objective okay. And a set of linear constraints right we already saw how to solve this you guys did right so you had a convex optimization tutorial. So one of the things that we are looking for from the convex optimization tutorial is that you will know how to solve this problem. so I say what we do after this, write the Lagrangian right.

(Refer Slide Time: 01:46)


s.t. $y_i(x_i^T\beta+\beta_0) \geq 1$
$\|\beta\| = 1/M$
$\min \frac{1}{2}\|\beta\|^2$
Lagrangian
$L_p = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i[y_i(x_i^T\beta+\beta_0) - 1]$

Fine, (1 – y) take the -1 is greater than equal to 0 when this is right. So right so I have to apply this for every data point so I run runs from I am I equal 1 okay. And I put a p there so so there is a primal so we will have to form the dual of this the dual looks a lot easier to solve okay duel is actually a lot easier to solve. So we will go ahead and do the duel, so for first I will set, I will take the derivatives right.

So derivative with respect to β and you can do that and solve it we will get that derivative with respect to β_0 so now is where I'm going to do some hand waving but you can go through this computation so take that substitute into this okay. And do a lot of simplification rights, so remember we have this β squared here therefore I am going to get a $\alpha_i \alpha_j y_i y_j$ kind of terms right.

So the duel will be so the duel is going to be a slightly simpler form why is it a slightly simpler form, so I have to only consider my constraints have become of lot simpler here right it just going to be α is should be non-negative that solves my constraints are so it turns out that there are efficient ways of solving optimization problems of this form right, you do not have to worry about it. Here are lots of packages that solve this seems for you.

But then Jesus you need to know what see optimization problem we are solving. I do not want you to use it as a black box. Essentially what you are going to be solving is this right. So when you have a solution, when you have something that is both primal and dual free so we can actually show that the duality gap is 0 in this case so it is not going to that. But the points when I have a solution to the problem right it has to satisfy certain conditions.

It is already looked at that the KKT conditions if people do not remember it please go back and revise that right. So there are a whole bunch of things so you need to for you need to have the solution to me primal feasible right. You need to have the solution to be dual feasible right and so that essentially have a bunch of things right. Primal feasible would mean that well your αi is have to be great that will be dual feasible way that will be one condition this
.

These need to whole right because it is a solution for the primal and there you are you have your complimentary slackness right. So that in this case becomes right. So, so I know if in the notes I think you saw it as λ IFI right. So this essentially that is it so this is α I into fi right. So this is this may affect so that is the fourth these are the KKT conditions, that need to be satisfied okay. And so what does this tell us?

This is a couple of things one so we know what the form of β should be what is the form of β it has to be α_i y_i x_i right. So it is essentially what you are going to do is your β will be taking out certain data points from your training data right and adding them up. So suitably have been multiplying it by the output the desired output, so if excise output was positive then this will be +1. If xi is output was negative this will be -1.

So it is going to take a few of those and they are going to add them up right. So this should remain you of perceptrons, so if you remember what we did in perceptions is we took whatever was misclassified we just kept adding it to the weight vector right. So in some sense you are doing something very similar to that but instead of having some kind of a heuristic approach to optimizing things right.

We did do a gradient descent right but then we just said ok we will arbitrarily pick the set of misclassified points and we will do the gradient descent and so on so for. But here we started off by saying okay we will minimize the distance to the closest point and from there we derive something and it looks very suspiciously like the perceptrons update rule okay. In fact nowadays when people say I am going to train a perceptrons.

They are actually doing this more often than using the perceptrons learning rule right way. So now something else that you can observe, so this condition has to be satisfied. This condition has to be satisfied. So let us look at it there are two terms here so when will this be 0when either this is 0 or that is 0 right. These are some condition when this has to be 0, sorry for what constraints okay fine but geometrically can you give me an answer.

Yeah! You are right but for geometrically can you give me an answer. So if this is when this has to be zero is when this guy is not 0 done right. So when will this gave me not 0, when it is not the closest point right. If if x_i is the closest point it will be bang on the margin right for a point here that term will be 0 right. For a point here that term will be greater than 1, right or appoint here the term will be greater than1.

You see that so since the term will be greater than 1 the term in the square brackets will be non 0 so α is have to be 0. Correct, so what does this mean it means that points that are further away from the hyper plane do not contribute to finding β. Because the α_s will be zero points that are far away from the hyper plane are not going to contribute in finding β. In fact the points that will contribute to β are exactly those points that are on the margin.

So in fact for this, this data set that they drew here right. Then only two important points at that one and this one, because only two points are on the margin right. That makes senses such points which lie on the margin are known as support points or support vectors right. And your β is going to depend only on the support points, what about β_0 okay. So we can plug in any data point here, and we can solve for β_0 right.

One of these support points you can plug it in here and you can solve for β_0 right. Which support point do you pick, ideally all of them should give you the same answer but usually does not happen because of numerical reasons. So what typically people do is they plug in all the support points okay. Solve for β_0 and take the average right. So each one in turn it for every support point you are going to get slightly different β_0 you just take the average okay.

So that is how you compute the hyper plane at the end of it is basically how here. Yeah p as it is that 50 it is potentially suppose so when would α be 0? If your data is on the hyper plane then on the merge sorry yeah! So that will be one case when that happens. Essentially you have two points which are on the same. It is not collinear but repeated things; I give you two data points that are on the same point right.

So by definition most of the support vectors will lie on the same line so it cannot be collinear okay. So right in such cases that could be the case but, yeah! These are generally degenerate cases yeah! So sure call them support vectors. If you want yeah! So one thing to note is my F hat right. So how this going to look like now that I given the form for β here. This is essentially going to look like I can flip these things around anyway that plus β_0 right.

So, so if you think about it I will come back to this point later so if you look at the duel I only have $X X^T X$ right and if you look at the final classifier I am going to use I am going to have $X^T X$ right so if i have a very efficient way of computing $X X^T X$ right I can do some tricks with this whole thing we will come back to that okay. I will just I want you to remember this so any questions on this, any questions on this?

So before we move on I just wanted to point out something so if you think about, how LDA works right. So LDA tries to do density estimation eventually right, if you if you think about it you make some assumptions about the probability distribution the form of the probability distribution. What assumption will you make; it is Gaussian with equal covariance across all the classes' right.

Though, that essentially means that every data point in your training set is going to contribute towards the parameters that you are estimating right. So the β will estimate there will depend on all the data points that were given to you, whether they are here right. Close to the hyper plane or whether they are very far away from the hyper plane. Let us all the data points will determine your class boundary, so that means that it becomes little susceptible to noise.

And if I have one or two data points that are generated through noise right even that will contribute to determining the separating plane hyper plane right. On the other hand we test with this kind of optimal hyper plane we are only worried about points that are close to the boundary right. So I can do whatever I want here right I can change move a few points over here and things like that it does not really matter.

What matters is if any noise enters close to the boundary right. So that so in some sense if my noise is uniform right the LDA will get more affected. Because even if noise insert some points there right LDA classifier will change right. Well my optimal hyper plane classifier will not move it will be affected only by that fraction of the noise that changes the actual decision surface right.

They make sense having said that I should point out that if, if your data is truly Gaussian with equal covariance LDA is actually optimal. It is probably optimal. While this one will depend on the actual data that you get but in general would say this is more preferable because this is more stable. People remember what stability is right; small changes in the data will not cause the classifier to change significantly right.

So here small changes in the data will not cause it to change significantly in an expected sense right. If I go and take the support vector and move it somewhere else okay. The class is the class boundary will change right. But then I have whole bunch of other vectors which I can move around nothing will happen to the class boundary unless I move it closer to the hyper plane than the existing support vectors right.

If I take a point from here and move it here of course the class boundary will change. As long as I do not modify which are the support vectors right I will get back the same classification surface again and again all right. So in that sense SVM or will come to SVM little bit, this kind of optimal hyper plane are very stable right okay. So any other questions this move on.

IIT Madras Production

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

www.nptel.ac.in