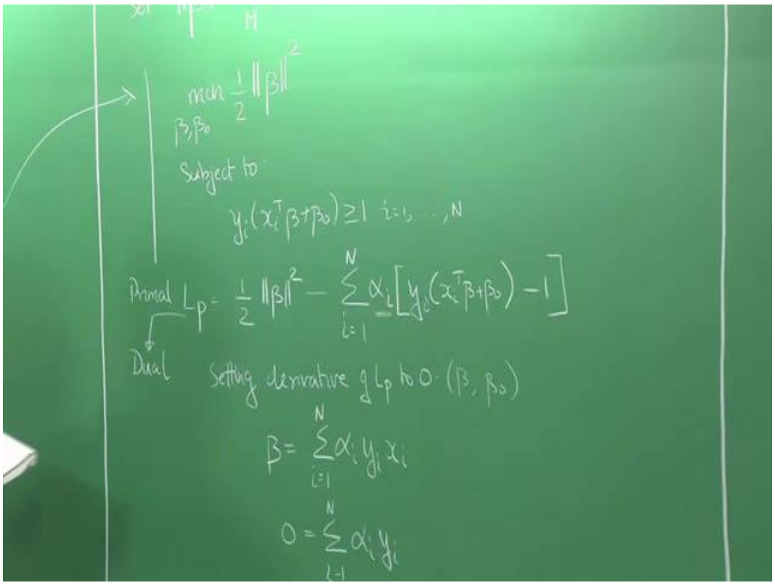
Introduction to Data Analytics

Prof. Nandan Sudarsanam
and Prof. B. Ravindran
Department of Management Studies
and Computer Science and Engineering
Indian Institute of Technology, Madras

Module – 05
Lecture – 30
Support Vector Machines

Hello and welcome back to our discussion on Support Vector Machines.

(Refer Slide Time: 00:17)



$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2$$

Subject to

$$y_i(x_i^T\beta+\beta_0) \geq 1 \quad i=1,\ldots,N$$

Primal $L_p = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N}\alpha_i\left[y_i(x_i^T\beta+\beta_0)-1\right]$

Dual

Setting derivative of L_p to 0 (β, β_0)

$$\beta = \sum_{i=1}^{N}\alpha_i y_i x_i$$

$$0 = \sum_{i=1}^{N}\alpha_i y_i$$

So, we were looking at the optimization problem corresponding to the optimal separating hyper plane. So, to solve this problem, so with one of the techniques for solving these kinds of constrained optimization problems is to set up a Lagrangian, which essentially looks at the original objective function, which is $\beta^2/2$ and the second component corresponding to the constraints that we have.

So, if you look at this quantity in the square brackets here, so you can see that this is the term on the left hand side of the inequality and that is the term on the right hand side of the inequality and we really want to make sure that, this difference is not negative. If this difference is negative, then that would mean that $((y_i * x_i^T \beta) + \beta_0) < 1$. So, we do not want this to be negative, so what we essentially say is, this we added here as with a minus sign.

So, this essentially means that, when I minimize this whole expression, so this term will become as large as possible, as largely positive as possible. So, that essentially means that I will go and try and make this as larger than 1 as possible. So, this term here α_i let us me control how much weight I want to give to satisfying the constraints versus how much I really want to minimize the objective function. So, we really need to satisfy the constraints as much as possible and since, there are solutions that will satisfy the constraint and give you good optima.

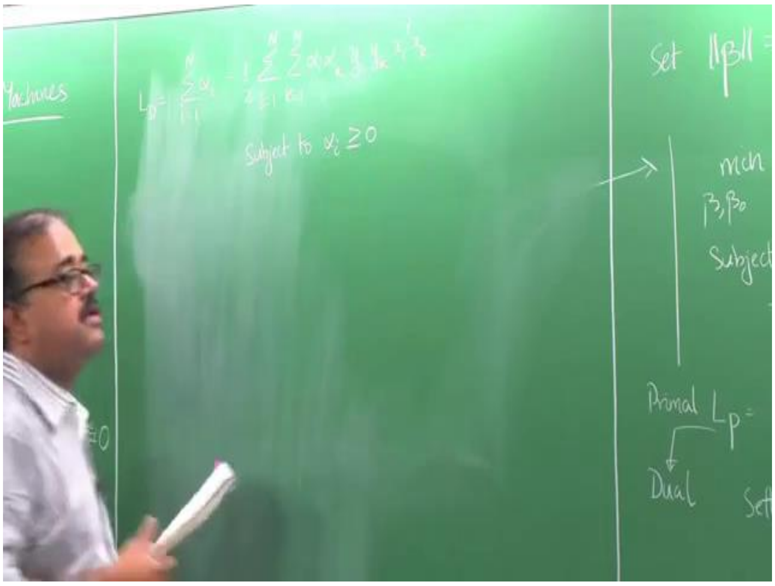
So, we should essentially be trying to derive this thing to as larger value as possible. So, this is called the primal of the problem and your goal is to minimize the primal. So, I am going to do something fairly technical right now. So, if you do not understand all of it in the first goal that is fine, you might have to do a little bit more reading on this side, but this is essentially give you an idea of how people go about solving these kinds of problems.

So, we are going to try and create, what is called the dual of this, the primal objective function. So, the dual is a way to create something that create an optimization problem, that is simpler to solve in some sense than the primal and the dual at all points provides you some kind of a lower bound on the kind of solutions that you can achieve with the primal and that the optima of the dual you ideally like the optima of the primal also to be achieved.

So, we are going to create a problem called the dual, we are going to solve the dual and when we reach the optima of the dual, you would like the optima of the primal to be also achieved. The same solution that gives you the optima in the dual problem should give you the optima and the primal problem and there are technical conditions under which this is satisfied and we are not going to go in to the technical conditions and this going to be give you a flavor of kind of results that will be looking at.

So, let us start by setting the derivative of L_p to 0, derivative with respect to β and β_0. So, taking the derivative with respect to $\beta = 0$ and solving for β gives me… So, you can figure there out by little bit of algebra here and likewise setting that derivative with respect to β_0 to 0 and solving it gives me. So, you can substitute these back into the primal problem and do a lot of algebra, do a lot of algebra really and then I can simplify this and I will get what is known as the dual, we write the dual here. So, this is just really obtained by substituting your β into the expressions here and then, using the fact that $\alpha_i y_i = 0$ at the optimum.

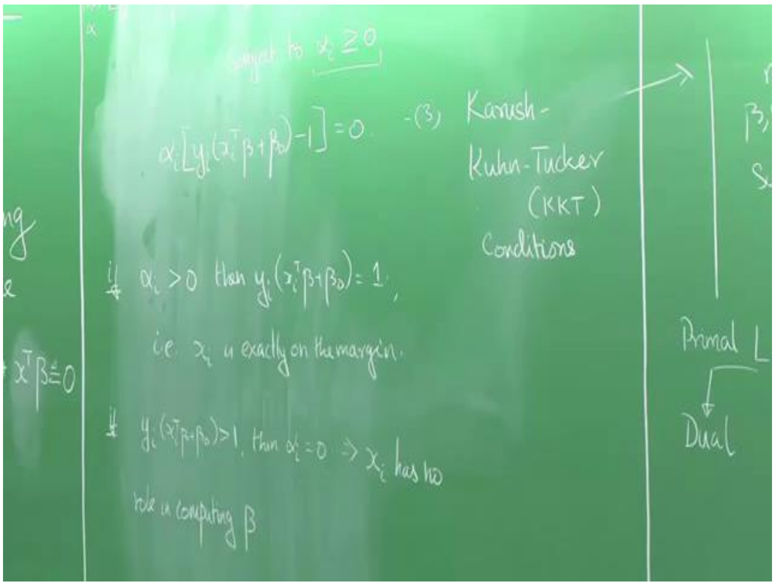
(Refer Slide Time: 06:35)



Machines
$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$
Subject to $\alpha_i \geq 0$
Set $\|\beta\| =$
min
β, β_0
Subject
Primal $L_P =$
Dual
Set

So, that is the thing, but then it is subject to be constrained. So, note that I said, so your dual is always going to give you a lower bound on the solution of the primal problem. So, really if you are minimizing the solution in your primal, it should be maximizing the solution in the dual, so that the two of them can coincide at some point. So, essentially you would be maximizing this subject to the constraint that, all your α_i's ≥ 0.

(Refer Slide Time: 07:21)



Subject to $\alpha_i \geq 0$
$\alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0$ -(3)
Karush-Kuhn-Tucker (KKT) Conditions
if $\alpha_i > 0$ then $y_i (x_i^T \beta + \beta_0) = 1$,
i.e. x_i is exactly on the margin.
if $y_i (x_i^T \beta + \beta_0) > 1$, then $\alpha_i = 0 \Rightarrow x_i$ has no role in computing β
$x^T \beta \leq 0$
Primal
Dual

So, if you think about it, this is the much easier constraint to wrap our heads around, because it just says that you will only be doing it in the positive co ordinates and while this had a more complex set of constraints. So, you kind of reduce the constraint to do something easier and therefore, the dual problem is sometimes easier to solve. So, for the dual and their primal to be at optima at the same time, so you really want them to satisfy a set of conditions, which are essentially to with the derivative of the primal problem.

So, we required that this should hold, we required that this should hold, we write them as 1, 2, 3. In addition, you required that this condition should also be required, that this condition should also be satisfied, these are called the KKT or the Karush Kuhn Tucker conditions. And so far, the optimization problem to have the same solution, we require that the KKT conditions should be satisfied.

So, once you have an optimal solution for the dual and the primal problem, because these KKT conditions have to be satisfied, you can make certain observations, especially we are working from condition 3 here. So, if $\alpha_i \geq 0$, so what does it mean. So, this has to be equal to 0, then the term in the square bracket has to be equal to 0; that means, $y_i * x_i^T \beta + \beta_0 = 1$. So, what does that mean?

It means that, it is exactly on the edge of the margin when it is equal to 1, because it is greater than equal to 1 is what we needed to satisfy, so when it is equal to 1; that means, it is exactly on the margin. Likewise if so, if the quantity in the square bracket is greater than 1, then α_i has to be 0, but that essentially means is, if your data point is something; that is far away from the hyper plane let us just more than the margin away from the hyper plane, then the corresponding α_is will become 0.

So, what does this mean for us, so if you think about it. So, the solution that we get, which is essentially β that is the solution that we want to get is formed by taking the product of α_i, y_i and x i. So, if saying the α_i is going to be 0, it essentially means that the corresponding x_i has no role to play in determining, what my β should be if I say that it implies if x_i is 0 that implies that x_i has no role in computing β.

So, which are the data points, which will actually effect the solution β here exactly those points for, which $(y_i * x_i^T \beta) + \beta_0 = 1$; that means, these are exactly the points, which lie on the margin. So, only these points will influence, how the solution β looks like and all the other data points that we have, which are farther away from the separately high per plane, then these points do not matter in the solution. So, these points are called support vectors.

So, you don't really have to solve this optimization problem yourself there are enough tools that actually can do it for you the whole goal of this lecture is to get you to appreciate, what is said that you are doing when you are using a support vector machine for solving a problem. So, at the end of the day all we are going to do is fire up tool that is going to tell you, what is the separating hyper plane given a bunch of data, But, it is good to have an appreciation of how the classifier is actually build.

So, once you figure out the β, then I can substitute that I can substitute that into the KKT the third condition here and solve for β_0. So, typically what you do is that you use every x_i that is a support vector and you substitute that here and then, try to solve for β_0 and typically end of taking the average value of that. So, the couple of things, which I want to point out about support vector machines.

So, one thing is we should be very clear that the training data none of the training data will fall within the margin, but that it is not to say that the test data might fall might not fall within the margin the test data might fall within the margin it might actually fall on the other side of the hyper plane. So, for all we know that the test data that could be errors on the test data it is just on the training data it tries to fix something there is as far away as possible from the data points.

So, the idea here is that, so if I give as much gap between the classes as possible, then the classifier would be more noise on either side. So, this is the assuming that the noise could be in this class or in this class if you know for sure that one class is noisier than the other or if one class is more valuable than the other. So, you might want to actually modify your objective, so that the line does not go write in the middle, but it is goes to one side or the other.

So, having said that under the assumptions of the support vector machines if assumptions hold good, then is a very, very robust classifier. So, the reason is it pays attention only to the points that are closest to the class boundary. So, you know I can have as many data points here I say want I can have as many data points here I say want of the corresponding class it will be does not affect my classification, because truly the once that are close the boundary are the once that need attention.

So, that essentially makes support vector machines more robust and on the other hand if you are going to have some kind of stochastic process that is generating the data right. So, if there are the few data points there are by chance or noise data points that actually close to the hyper plane that will affect the support vector machines tremendously. And therefore, it will try to reduce the margin by a large extent while classifier that looks at the entire data and tries to find the distribution for the entire data might be a little bit more robust to this kinds of noise. So, this is the, this is how you solve the basic optimization problem for support vector machines.