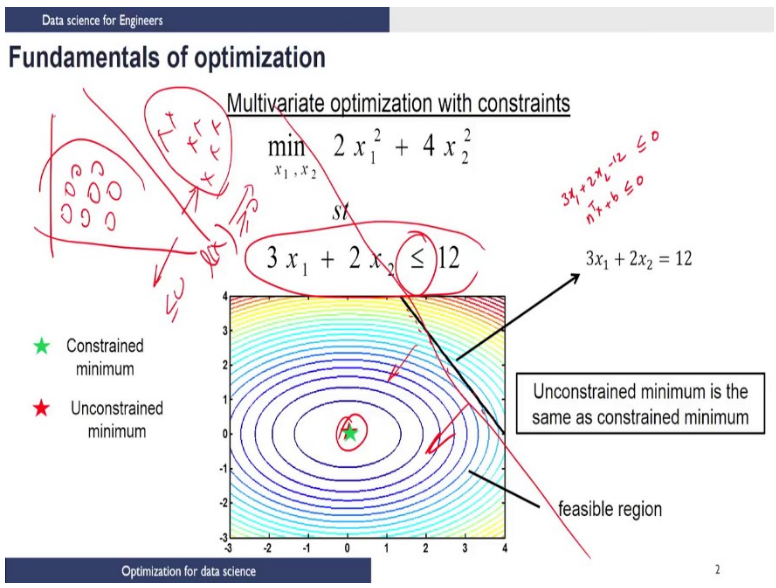
Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer and Science Engineering
Indian Institute of Technology, Madras

Lecture – 28
Multivariate Optimization with Inequality Constraints

We come to the last topic in this series of lectures on optimization for data science. Till now we saw how to solve unconstrained multivariate optimization problems. Then we saw how to solve multivariate optimization problems with equality constraints. Now we move on to multivariate optimization problems with inequality constraints. Before I start explaining some of the key ideas in these types of problems, let us look at why we might need this optimization technique in data science.

Remember in one of the earlier lectures I talked about data for people who like south Indian restaurants and so on.

(Refer Slide Time: 01:07)



So, if you remember that example I said there are a group of people who might like certain type of restaurant there might be a group of people who do not like that kind of restaurant and so on. Then if we were to do a classifier which probably is a line like this then, as we are trying to solve an optimization problem to identify a classifier like this, we have to impose the constraint that all these data points will have to be on one side of the line and all of these data points will have to be on the other side of line and from our lecture on half spaces and hyper planes in linear algebra we know that if this equation is something like this which is linear equation, then it might be that if the normal is in this direction then this direction is said that if I substitute a value of this point into this it is greater than equal to 0 and on this side it is less than equal to 0 with 0 being the line.

So, now, notice that for each point if we were to write the condition in terms of the equation of the line and then you would see that these become inequality constraints. So, there may be as many in equality constraints there are points and so on.

So, you see why we might be interested in imposing inequality constraints and optimization problems from a data science viewpoint. A more sophisticated version of this idea is what is used in one of the data science algorithms called support vector machine. Though we will not study that technique in this first course on data science for engineers, I just wanted to point out that this class of optimization problems is very important from a data science viewpoint.

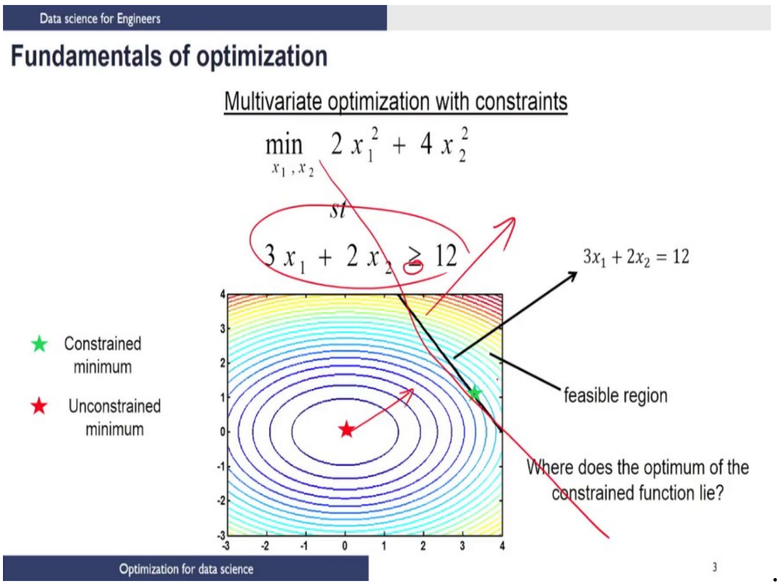
Now, let us go back to the same example that we had before, where we had the equality constraint and then we tried to solve the optimization problem and then just make that equality into an inequality. So, in this case let us assume what was equal to 12 in the last lecture has now become less than equal to 12 and let us understand intuitively what happens to problems this of this type. Now in the previous case we said when we have an equality constraint we said we are interested in any point on this line as a candidate solution these are all called the feasible points and of all of these points we were trying to pick the point which will give me the minimum objective function value.

Now, when I have the optimization problem where I have this inequality and let us assume this is less than equal to in this picture what we have plotted is that the original unconstrained optimum or the unconstrained minimum is still the red star. Now if you look at this and then say this less than equal to 12 it can be basically rewritten as $3 x_1 + 2 x_2 - 12 < = 0$ and remember that this is of the form in transpose x + b < = 0. So, it is going to be one half space depending on how n is defined. In this case it will turn out that this is the half space that is represented by this equation.

So, the difference between the equality constraint and the inequality constraint is the following, in the equality constraint we had every point on this line being a feasible solution when you make this as a inequality constraint. Then what happens is if you think of this line is extended all the way, any point to this half space now becomes a feasible solution and because every point in this half space is a feasible solution the unconstrained minimum also becomes a feasible point so the constrained minimum and unconstrained minimum are the same so this is an interesting thing to see.

So we saw that in the case of equality the unconstrained and constrained minimum were different, but when we made this into an inequality with the less than equal to sign we see that the constrained and unconstrained minimum are the same points.

(Refer Slide Time: 05:38)

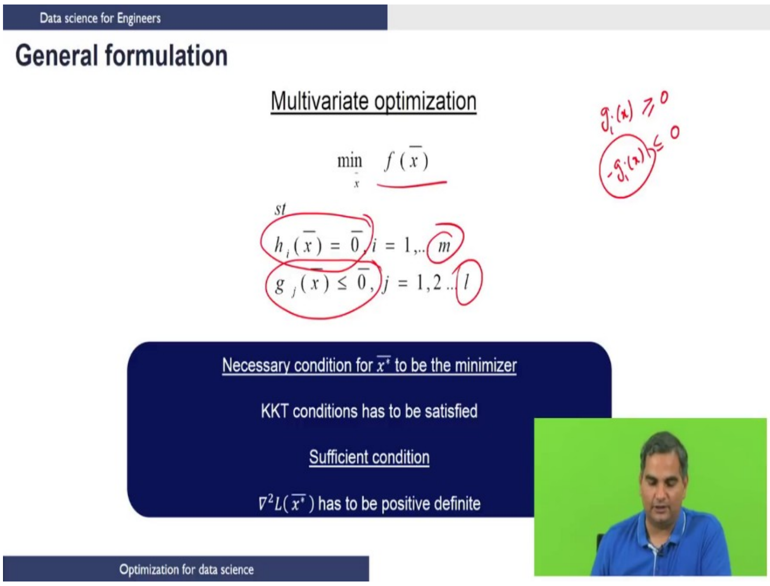


Now, let us try and see what happens if I flip the sign and then said this is greater than equal to 12. Then what would happen is if you were to extend this line all the way, then the feasible region is to this side. Now, you asked the question where does the optimum lie? You will notice that again we know the best solution is here and as we move away from this solution, we will see that we are losing out on the objective function value and as before we know any point on this line or to the side is a feasible point and any point on this line is also feasible because of this equality sign.

Now, making the same arguments that we made in the case of the equality constraint problem, we will see that I give up on my optimality, that is I keep going through contours of larger and larger size where the optimum value keeps increasing and when a contour particular contour touches this line exactly at one point then I have a feasible point which is going to satisfy this constraint, the equality part of the constraint. So, it is satisfying the general constraint and that is the worst I have lost in terms of how much my objective function has increased its value by.

Anything more would be unnecessary because if I move little further I am going to make my objective function value worse. However, there is no need to do that because I already found a feasible point here. So, this case the constrained minimum becomes the same as the minimum that we achieved with the equality constraint case. So, you see that depending on what type of inequality these different things can happen.

(Refer Slide Time: 07:46)

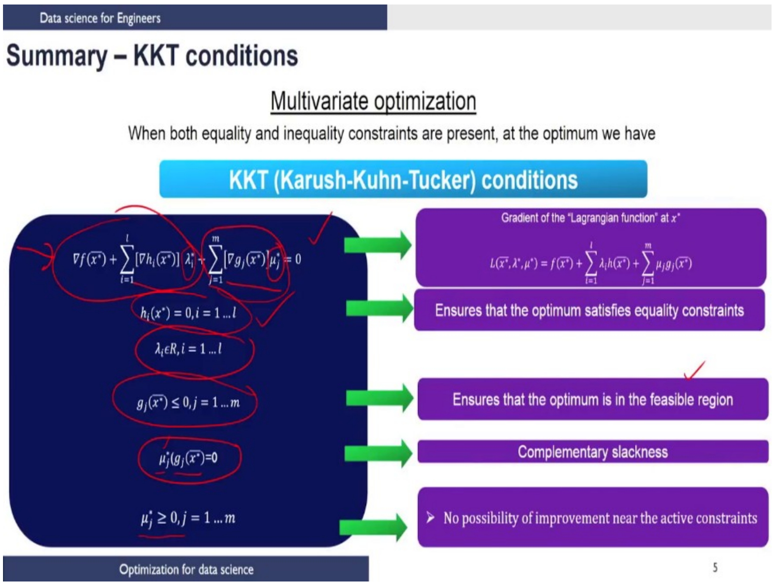


Now, from a general mathematical formulation viewpoint, we are going to generalize this. What I am going to show is I am going to show you the general conditions. Now for this course we might not be using this later in any data science algorithm. Nonetheless this forms a basic idea for more sophisticated data science algorithms like s v m and so on. So, it is worthwhile to pay attention and then understand this completely. So, general multivariate optimization problem we can say has both equality and inequality constraints.

So, I have the objective function, I have m equality constraints and l inequality constraints. Notice that we can always write the equality condition in this form where I have 0 on the right hand side because if you have something on the right hand side I simply move it to the left hand side and get a 0 on the right hand side. Similarly, any condition inequality condition whether it is greater than equal to 0 or less than equal to 0 I can always put it in this form. If the original condition is already in less than equal to form then it is the same as this if it is in the greater than equal to form then what I do is I multiply by a negative and then I have this condition has less than equal to 0.

So, now, I call this my constraint so I can again put it in this form. Now this becomes a little more complicated formulation and I am going to show you the conditions for this in the next slide. Which will look a little complicated in terms of all the math that is there. What I am going to do is I am just going to simply read out the conditions in the next slide and then we will take a particular example and then demonstrate how the conditions work.

(Refer Slide Time: 09:45)



These are the conditions for multivariate optimization problem with both equality and inequality constraints to be at it is optimum value and let us look at this carefully. Remember when there was only equality constraints, we had this part of the expression where we had ∇ of f + linear sum of λ times ∇ h for each one of the equality constraints.

Now, when you also have inequality constraints, what happens to this equation is, just like how we added a single parameter for every equality constraint, we add a parameter for each inequality constraints. So, if there are m inequality constraints there will be m parameters and in typical optimization books and literature people use λ as a scalar multiple for each one of these constraints. So, if there are l equality constraints, there will be l lambdas and people use the nomenclature of μ for the inequality constraints. So, if we have m μ s there will be, there will be m μ s corresponding to the m inequality constraints.

So, the difference between this condition in the equality and inequality case is that for every one of these inequality constraints you add more linear combinations of μ times δ g. So, look at this, this is the same form of as this except that I used λ here and μ here, but I take a take a ∇ of h and $\nabla$of g. That is the first set of conditions then much like how we had the constraints equality constraints also as part of conditions in the previous case, I am going to have the optimum solution satisfy all of this equality constraints.

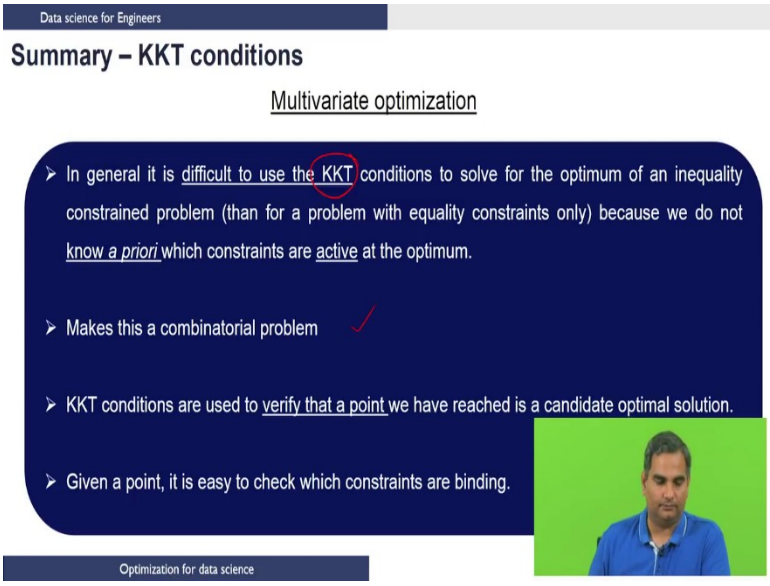
Now the λ is some real number, so as many real numbers as there are equality constraints and much like how I still need to have this equality condition satisfied the optimum point, I need to have the inequality constraint also to be satisfied by the optimum point. So this ensures that the optimum point is in the feasible region. Now this real differences between the equality constraint condition and the inequality constrained situation shows up. We also have additional constraints, which are of this form, these are called complementary slackness condition.

So, what this says is if you take a product of the inequality constraint and the corresponding μi then that has to be 0. Basically what it means is either μj is 0 in which case this is free to be any value such that this condition is satisfied or this is 0 in which case I have to compute a μ and the μ that I compute has to be such that it is a positive number or it is greater than equal to 0. So, this condition is there to ensure that whatever optimum point that you have, there is no possibility of improvement -any more improvement- from the optimum point so that is the reason why this condition is there.

Now just keep in mind that if you are seeing course on optimization for the first time it is not very easy or natural to understand this constraints right away. However, what we are going to do is in the next slide we will take an example and then show you how these things work. One thing that I want you to keep in mind is if we had let us say an unconstrained optimization problem objective function in n variables, I always look at whether the optimum conditions have enough equations and variables for me to be able to solve the system of equations and clearly you know in the unconstrained case you have n equations and n variables and I clearly made the point in the equality constraint case that for every equality constraint you add an extra parameter.

However, you will have enough equations and variables because when you write the first condition which is of this form you will get the n equations and you will get as many equality constraints that need to be satisfied as there are lambdas. So that also works out properly. Now we will see whether the same thing happens in the case of inequality constraints.

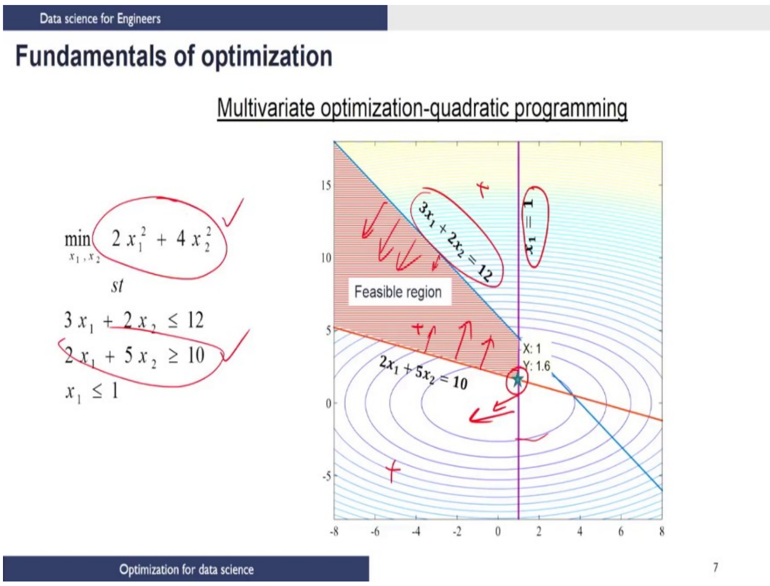
(Refer Slide Time: 15:02)



There is a specific problem in solving this -what are called the KKT conditions-the conditions that I showed in the previous slide are called the KKT conditions. It is not easy to solve the KKT conditions directly in the inequality case because of the complimentary slackness condition which says either μ could be 0 or z could be 0. So, we have to make a choice as to which is 0 so that makes this a combinatorial problem.

So, in general the KKT conditions are generally used to verify if a solution that we have is an optimal solution. However, optimization algorithms will have different ways of solving this problem and there are methods called penalty methods, active set methods and so on, we are not going to talk about those methods here. Nonetheless I just want you to understand how the solution works out in an analytical case.

(Refer Slide Time: 16:07)



Let us take a look at a numerical example to bring together all the ideas that we have described till now. In this particular case it is a multivariate optimization problem which is actually called quadratic programming and this is called quadratic programming because the objective function is quadratic and the constraints are linear. Those types of problems are called quadratic programming problems.

I think this is the same objective that we have been using till now in the several examples. So, let us say this is the objective function and let us assume that we have constraints of the form shown here. Let us assume the first constraint is $3 x_1 + 2 x_2 <= 12$, the second constraint is $2 x_1 + 5 x_2 >= 10$ and the third constraint is $x_1 <= 1$. Now I am not doing anything more to this problem, but nonetheless I just want you to remember that to be consistent with whatever we have been saying this should be converted to a less than equal to constraint which we will see how that happens in the next slide.

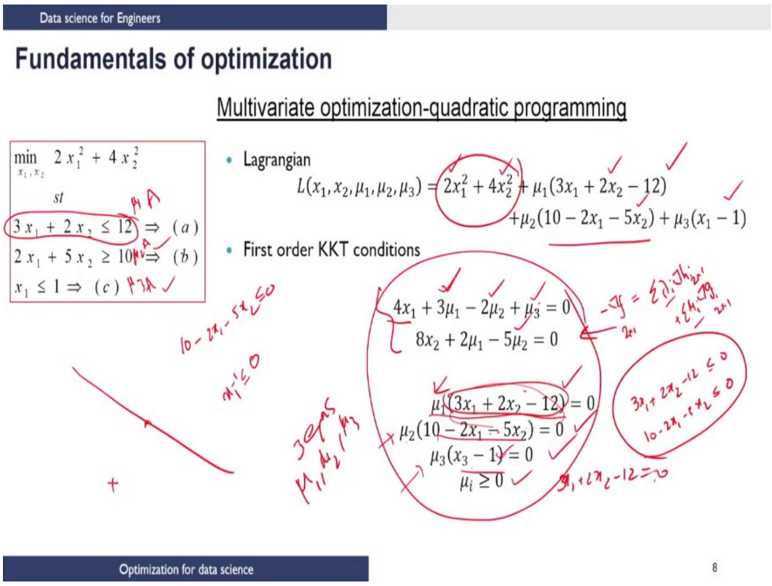
Now let us look at this pictorially. Remember that the value of the objective function is going to be plotted in the z direction coming out of the plane of the screen that you are seeing. So, the representation of that are these contours that we see here, which are constant objective function contours. So, I am just trying to see and explain how this picture speaks to both the objective function and the constraints. So, the objective function is actually represented in this picture as this constant value contours. So, if I am moving on this the objective function value the same we have repeated this several times.

Now in this x_1 x_2 plane, let us look at how these constraints look. So, I have this equation which is the equation of a line. So, when we talk about the first constraint which is less than equal to 12, then any point to this side of the line is a feasible point. Now, when we look at this constraint here then the equation of the line is $2 x_1 + 5 x_2 = 10$ and whenever I have something greater than = 10, this region is a feasible region and this is the $x_1 = 1$ line and $x_1 <= 1$ would be this region.

So, if you put all of these regions together the only region which is feasible is shaded in brown colour here. So, if you take a point any point in this region it will be satisfying this constraint because it is to this side of this line, it will satisfy the second constraint because it is to the side of the line and it will satisfy the third constraint because it is to this side of the line. Notice that if you take any point anywhere else you will not be feasible. For example, a point here would violate constraint 1, but it would be feasible from constraint 2 and 3 viewpoint nonetheless all the constraints have to be satisfied. Now similarly if you take a point here while it satisfies constraint 1 and 3 it will violate constraint 2.

Now, if you notice the optimum point is going to lie here and we are going to try and find out this value through the conditions that we described in the last few slides.

(Refer Slide Time: 20:10)



So, what we do here is the following. So, we de ne something called a Lagrangian, which basically will boil down to the kind of conditions that we have been talking about I will explain this quickly. So, we take the objective function value or the objective function expression here and then there are no equality constraints here, there are only inequality constraints. I said we had one parameter age for each of these inequality constraints.

So, this is already in a form that is palatable to us which is less than equal to form. So, I just simply write this as μ one times 3 x_1 + 2 x_2 - So, 3 x_1 - + 2 x_2 - 12 < = 0. So, this is already in the less than = form, when I look at the second equation, I want to make it into a less than = form then what I do is I said 10 - 2 x_1 - 5 x_2 < = 0. So, if I put this here this is into the less than = form and then corresponding to this if I have μ 1 corresponding to this μ2 and corresponding to this I have μ3 I have x_1 - 1 < = 0 so you see that term here.

Then what you do is, you differentiate this with respect to x_1 and x_2. You will get the first 2 conditions that we have been talking about for the con-strained case with equality constraint, unconstrained case and so on. So, the 2 equations for the 2 decision variables so, when you differentiate this whole expression with respect to x_1 4 x_1 comes out of this term right here and this is only a function of x_2. So, differential with respect to x_1 will become 0 and I will have 3 μ 1 x_1 when I differentiate this with respect to x_1 I get 3 μ1 and from the second term I will get - 2 μ2 and from the third term I will get μ3. So, th= 0. So basically this equation you have and then when I differentiate the same expression with respect to x_2.

So, I get 8 x_2 from this here and from the second term I get 2 μ1, I get - 5 μ2 from this term and from this term I get nothing because it is only a function of x_1. So, I get this equal to 0 so, this basically gives you the condition that that we have talked about which is of the form in the equality constraint case remember I said you have - ∇ has to be = σ λ I ∇of hi and in the inequality case you also add + σ μi ∇of gi if gi are the inequality constraints.

So, you kind of back out these 2 equations from this constraint and you have 2 equations here because ∇ of x is of size 2 by 1 h is 2 by 1 2 by 1 because there are 2 decision variables and these are scalars and this is a linear weighted sum. So, if you notice all of this, you see that I have 2 equations. However, I have 5 variables that I need to compute, I need to compute a value for x_1, I need to compute a value for x_2 and then I need to compute μ1, μ2 and μ3. So, let us see how we do that. We go back and add the complementary slackness conditions. Also keep in mind that other than this we also have to make sure that whatever solution we get still has to satisfy the 2 inequality constraints also.

So, whatever solution we get should still be such that 2 x_3 x + 2 x_2 - 12 is less than = 0 and 10 - 2 x - 5 x_2 is less than = 0. So, these 2 also have to be valid, but because these are inequality constraints we cannot actually use them to solve for anything. Once we solve for these variables we have to still verify whether this conditions are satisfy or not. So, going back to the complementary slackness condition we saw that we will have μ times g is 0. So, this is corresponding to the first inequality constraint, this is corresponding to the second inequality constraint and this is corresponding to the third inequality constraints and we also have to have this μi greater than = 0. So, all of these conditions have to be satisfied for an optimum point.

Now, it is interesting to notice something here for let us just take this as an example already we have 2 equations, I have already mentioned that the 2 equations are here. So, I am looking for another 3 equations to compute μ1, μ2, and μ3. So, that will be a total of 5 equations and 5 variables. So, for this to be 0 you could say in this equation either μ1 is 0 or whatever is inside the bracket is 0. So, we could say one possibility is whatever is inside the bracket is not 0 in which case μ1 has to be 0. So, the key point that I want to make here is if we say whatever is in the bracket is not 0, then that automatically gives me the value for μ1. So, out of the 5 variables here I have already computed one, similarly if I say whatever is inside here is not 0 then μ 2 has to be 0 and whatever is inside here is not 0 μ3 has to be 0 then I have already computed values for μ1, μ2, μ3.

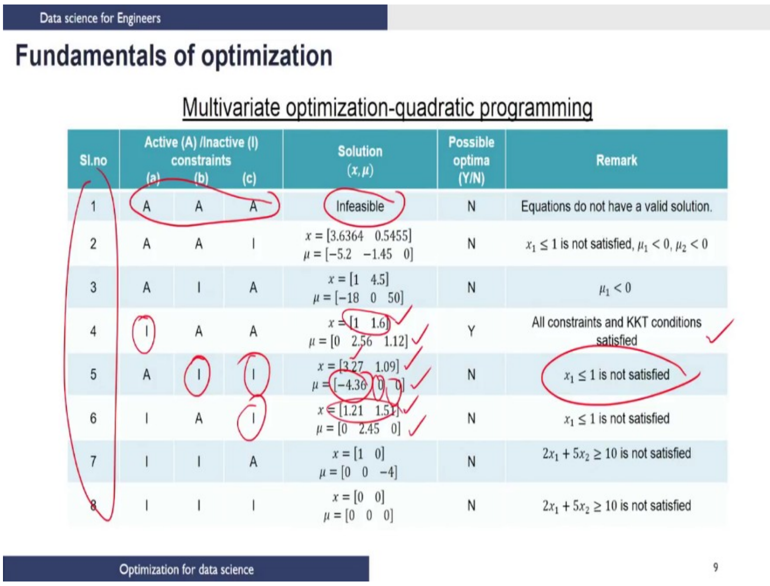
Then I could substitute those values back into this equation and I have 2 equations I can calculate x_1 and x_2. So, this is one option. So, in one of the previous slides I had mentioned that this becomes a combinatorial problem because we could also assume that this goes to 0. So, this term goes to 0 and now let us assume actually this is nonzero, this is nonzero, let us see what happens to that case do we have enough equations and variables.

So, in this case we will have 1 equation, 2 equations, the third equation will be μ2 has to be 0 because we have assumed this is not 0 the fourth equa-tion has to be μ3 is 0 because we have assumed this is not 0. Now the fth equation becomes the one which we have assumed which is the term inside the bracket is 0. So, in which case again I have 1 2 3 and then μ2 = 0 μ3 = 0 as 5 equations in 5 variables. You could for example, assume that this and this are 0 in which case I have to compute μ2 and μ1 and then let us say this is not 0 then μ3 is 0, but in that case also you will have 1 equation, 2 equation th= 0 is 1 equation and th= 0 is 1 equation. So, again I have 5 equations and 5 variables

So, whichever assumption you make as far as these equations are concerned you will have enough equations and enough variables. However, for some of those choices when you actually find a solution and try to plug this back into this right here it might not satisfy this. So, in which case that is not a viable option for us from an optimization viewpoint. And in some cases this might be satisfied, but the μ that you calculate out of the equations you get might not be positive. So, it might get negative μ in which case again this is not an optimum solution. So, let us see how this happens for this example.

Now, I am going to use one notation here so, that we can understand the table in the next line. So, whenever I assume that an equality constraint is exactly satisfied; that means, when I say $3 x_1 + 2 x_2 - 12 = 0$, then we say this constraint is active it is active because the point is already on the constraint. If I take a point here then that is not on the line so, that is basically less than = 0 so, I will say it is inactive. So, for every constraint I can say whether it is active or inactive. So, if this constraint is active; that means, $3 x_1 + 2 x_2 - 12$ is 0. If this constraint is active; that means, $2 x_1 + 5 x_2 - 10 = 0$ and if this constraint is active; that means, $x_1 = 1$.

(Refer Slide Time: 29:38)



So, in the example that we are considering right now, there are 3 inequality constraints and as I mentioned in the previous slide if the inequality constraint is exactly satisfied that does it becomes an equality constraint then we call that an active constraint and if the constraint is not exactly satisfied we call it as an inactive constraint. And now we have 3 inequality constraints and each of these constraints could be either active or inactive.

So, there are 2 possibilities for each of these constraints and since we have 3 constraints there are 2 to the power 3 possibilities which = the 8 possibilities that we have here. So, what I am going to do in this case is we are going to enumerate all possibilities for you to get a good understanding of how this approach works when you have inequality constraints. So, let me pick let us say a couple of rows from this table to explain the ideas behind how this works and what we are going to do is in the next slide we are actually going to see graphically what each of this case means.

So, let us look at the first row for example, here the choice we have made is all the 3 inequality constraints are active. That means, they all become equality constraints. Notice something interesting remember there are 2 decision variables. So, each inequality constraint is basically representing one half space for a line and when they become active each of these constraints become an equality constraint each of them become line.

Now when all 3 are active then we have 3 equations in 2 dimensions right. So, there are only decision variables x_1 and x_2, but I have 3 equations in those 2 variables and from our linear algebra lectures we know that when we have more equations than variables in many cases we are not going to find a solution for the 2 variables which will satisfy all the 3 equations. So, in this case you cannot solve this problem it is infeasible because though I have enough equations a subset of these equations 3 equations are in 2 variables and I cannot find a valid solution. We will understand what this is geometrically in the next slide.

Let us look at some other condition here. Let us pick for example, row 5. So, if you look at row 5, we have made a choice that the first constraint is active, the second constraint is inactive and the third constraint is inactive, that basically means the first constraint equal to 0, the second and third constraints have to be less than equal to 0, which needs to be tested after we go through the solution process. Now much like how I described before in this case also we will be able to find 5 equations and 5 variables and we can solve for x_1 and x_2 which is shown here and we have solved for μ which is shown here.

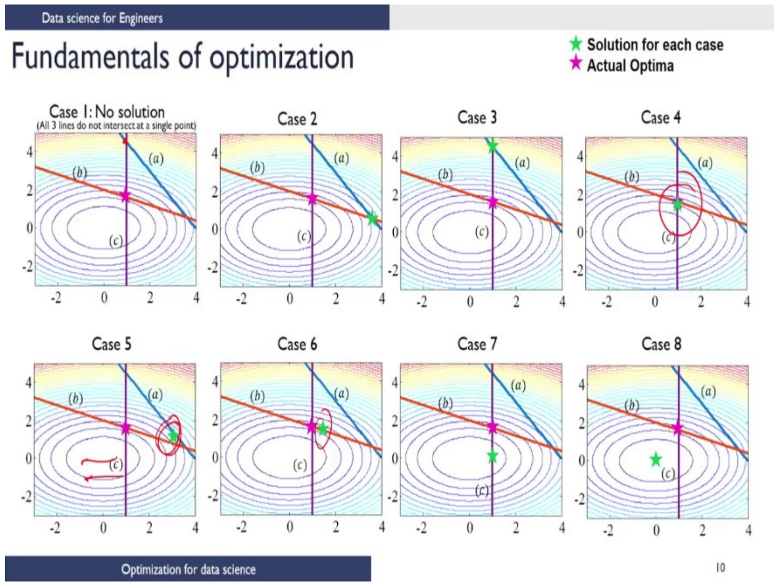
Now if you just look at this solution, you will you will say it seems to satisfy everything because I have a solution for x_1 and x_2 and I have $\mu 1$, $\mu 2$, $\mu 3$ where $\mu 1$ is - 4.36 and $\mu 2$ and $\mu 3$ are 0. However, when you look at this μ you will see that one of the μs is negative. Which basically means that this cannot be an optimum point based on the conditions we showed in a couple of slides back, not only that on top of it when you actually put these 2 values into the constraint x_1 is less than equal to 1 it is not satisfied because x_1 is 3.27.

So, if you take this row for example, it looks like both this μ being positive is not satisfied and this constraint is also not satisfied. An interesting thing to note here is we have to go back and check only the constraints that we have assumed to be inactive because the active constraint is already at 0. So, whatever solution again will automatically satisfy the active constraint. Let us say row 6 for example, here we have made a choice that the first constraint is inactive, the second constraint is active and the third constraint is inactive.

Again we will get 5 equations and 5 variables and we will look at the solution here. It again looks good from the x view point. We have solved for x_1 and x_2, 1.21 and 1.51. Unlike the previous case in this case the μs also looked good because μs have to be positive. So, I have 0 to 0.450. So, it looks like this, this is a good candidate for an optimum point. But once you have satisfied all of this you have to still go back and look at whether the inactive constraints are satisfied by this solution point right here and the inactive constraint here is x_1 has to be less than equal to 1, but if you notice that 1.21 does not satisfy this constraint. So, this cannot be an optimum either.

When you look at row 4, where we have assumed constraint 1 is inactive and 2 and 3 are active, I get a solution x_1 1.6. I get all μ s to be positive which is also one of the conditions for an optimum point and then I have to only verify this constraint here because other 2 are active constraints and they are providing equations for us to solve so they are like they are going to be valid. Now when you put these 2 values into the first constraint you will check that it actually satisfies the inequality also. So, this is a case where all constraints and KKT conditions are satisfied. So, this is the optimum point so, the optimum solution is 1.6 which is what we had indicated in the slide with the geometry of this problem.

(Refer Slide Time: 36:56)



So, let us look at each of this case in terms of where the optimum lies and how we interpret this. So, if you take the case one where we took all the constraints to be active we said we have 2 variables and 3 equations and that is not satisfiable in general. Geometrically what this means is we want the point to lie on all the 3 lines at the same time because we have assumed all 3 lines are active concerned. That means, all of them have to be equal to 0.

So, you find that you cannot get a point where the all 3 lines equations are satisfied. So, if I want to satisfy 2, I will get a point here nonetheless it would not satisfy the other constraint and so, on. So, this is the case of not all 3 lines intersect so we do not have a solution here. Similarly you can look at each of these cases and you can see what happens in each of these and you will notice that only in case 4 will you have the solution that you got from the Kuhn Tucker condition and the actual optimum being satisfied.

In every other case you will see that there is some problem or other. So, if you go back I think we looked at case 5 and 6 then the solution for case 5 is this, the solution for case 6 is this and we said these two are not good solutions because they violate the condition x_1 is less than equal to 1 which basically seen here because the feasible region is to this side of line x_1 is less than equal 1, but this point is violating this constraint and here again you see that this point is violating this constraint.

So, in summary what you need to know is that in the unconstrained case it is very clear that I just simply write this ∇ of f is 0 as the condition and I will have enough equations and variables there will be n equations and n variables. In the constraint case with equality constraints I will get the same n equations, but the form will be slightly different we write that as - ∇ f is σ λi ∇ of hi and since we add as many variables as there are equality constraints and since the equality constraints have to be satisfied those give you the extra equation. So, you will have enough equations and variables.

In the inequality constraint case, the first n equations come from a very similar form where - ∇ f is σ λi ∇ of hi + σ μi ∇ of gi that gives you n equations and corresponding to every one of those λ corresponding to equality constraints you will get so many equations which are the equalities have to be satisfied. The only complication comes in when you have these inequality constraints where either you know you have can have one or the other be 0 that is what we call as complementary slackness.

So, we have this μ times g going to be 0. So, if g is 0 we call that as an active constraint in which case we have to calculate the μ corresponding to it and in the optimum point μ will be greater than equal to 0. Now depending on the form in which you write whether you write all of these constraints or as inequality constraints less than equal to or greater than equal to and also in terms of whether you write the original equation as - ∇ f equal to this sum on the right hand side or ∇ of equal to sum on the right hand side the sine of μ in different textbooks and different papers might be reported as either they have to be positive or they have to be negative.

So, you have to be careful about the conditions when you look at those, but if you stick to this type of writing the equations, where you write - ∇ of a σ λ ∇ hi + σ μ ∇ gi and if you write all the constraints as less than equal to inequality constraints, then μ s have to be positive for the point to be an optimum point which is what we saw here.

As I mentioned before while this unconstrained case is a very very important case in machine learning algorithms such as s v m and so on the constrained case I mean we might not be using this quite a bit in this course. Nonetheless for the sake of completeness and for the sake of giving the foundations for understanding other data science algorithms that outside of this course that you might go and study.

I have also described the key ideas behind how to solve constrained optimization problems when you have equality and inequality constraints. Keep in mind that while I have shown you the conditions, I have not shown a proof or a derivation of these conditions in a formal manner. The equality constraint case I appealed to intuition to tell you why the conditions turn out to be the way they are. However, all of this can be formally proved and you can derive these conditions based on formal mathematical arguments.

So, with this we conclude this portion on optimization for data science now we have all the tools that we need to understand data science problems. The next set of lectures what I am going to do is I am going to introduce to you different types of data science problems that we encounter, how do we think about these data science problems. Is there some formal way of thinking about these problems; that we can use to solve a variety of problems and then we will move on to regression as a function approximation tool and we will look at different clustering techniques that are used in data science and then we will finally conclude with one particular technique called principle component analysis which is very useful for engineers and end with more general example of how one solves real life data science problems.

So, I hope to see you continue these lectures and understand more of data science.

Thank you.