Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematic

Institute of IIT Kanpur

Lecture: 40

Hello everyone, this is lecture 5 of week 8. In the previous lectures of this week, we learned in detail about the augmented Lagrangian method. We learned the algorithm step-by-step for problems with only equality constraints, and then we learned the algorithm step-wise for problems with both equality and inequality constraints. We also saw quite a few examples. In the previous lecture, we saw a practical example, which was the consumer utility maximization problem. Towards the end, I mentioned that both of these methods—the quadratic penalty method and the augmented Lagrangian method—are used only for non-linear programming problems and not for linear programming problems. So why is that the case? We will discuss that in this lecture.

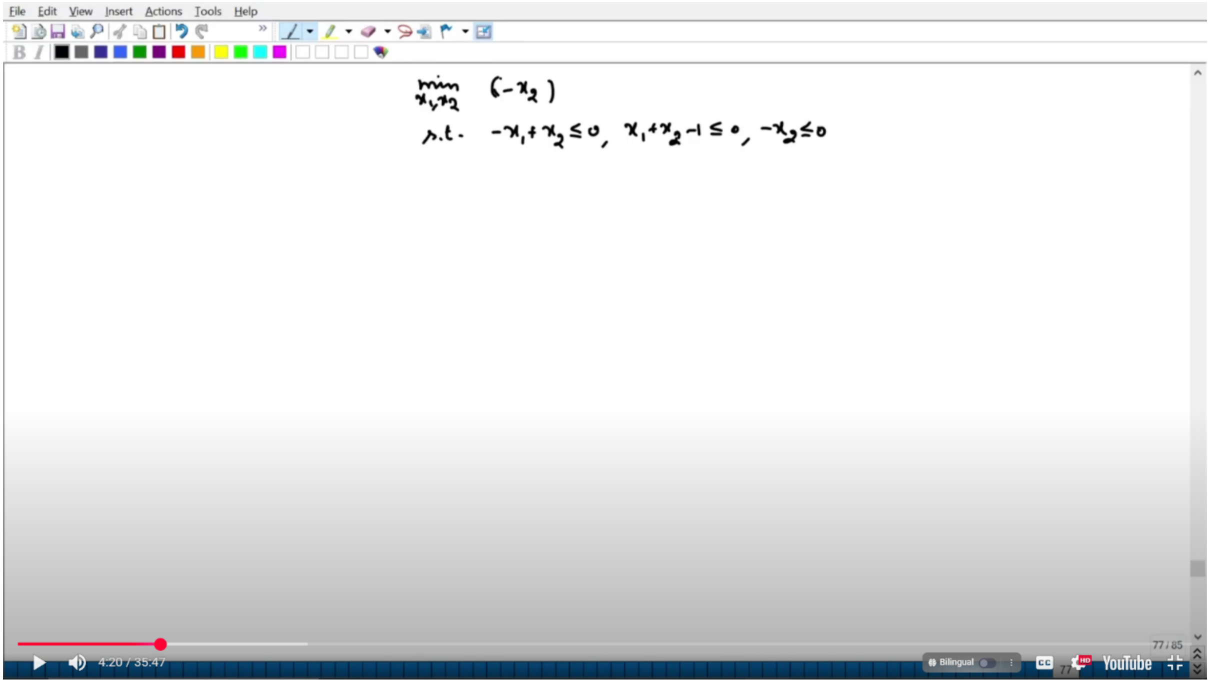
We will first start with an example. If you look at the examples we have worked out, you can see that in a particular example the constraints are linear, but the objective function is non-linear. I can take the same example but change the objective function to a linear objective function. If I do that, I would clearly have a linear programming problem. In a similar way, if you look at the consumer utility maximization problem, you can see that the constraints there are linear as well.

So, if I modify the objective function to be linear, we will have a linear programming problem. I will do that for this particular problem where we have the triangle with vertices (0,0), (1,0), and (½,½) as the constraint set. Instead of minimizing the distance from the point (3,2) to the feasible set, we will write down a linear objective function.

I am going to write down the objective function to be $-x_2$. That is, I am going to maximize x_2 subject to the condition that the feasible set is this triangle. Just by looking at the figure, you can figure out that if you want to maximize x_2, the answer is ½, because there is a vertex at (½,½).

So if I write down the problem to minimize $-x_2$ (since maximizing x_2 is just minimizing $-x_2$) subject to the same set of conditions $-x_1 + x_2 \leq 0$, $x_1 + x_2 - 1 \leq 0$, and $-x_2 \leq 0$, the answer is -½.

(Refer Slide Time 4:20)



Now, let us check if we give this particular problem to the code that we already have. We have already written the codes for augmented Lagrangian methods for various constraints. For this particular constraint set, we have already written down the code. So let us actually use that particular code and check if it gives us the right answer.

This is the augmented Lagrangian method for utility maximization, which was the problem where we minimized $\frac{1}{2}(x_0 - 3)^2 + (x_1 - 2)^2$ subject to the feasible set being the triangle.

I am just going to change f(x) to $-x_1$.

I want to retain the original problem, which is why I am commenting it out instead of removing it. Similarly here, I will comment this out and write a new line of code, again to retain the code that is already there. So if f(x) is $-x_1$, then the gradient will be (0, -1), and for the Hessian, we just have to remove the previous terms.

I have just commented out the relevant part. Let us choose the initial point to be (3, 2), which is some point outside the feasible set. Let us choose μ to be (0, 0, 0) as we usually do. Let us check what happens with this code. Does it give the right answer or not? It gives an error, saying that we have got a singular matrix as the Hessian. The code says the Hessian is singular.

In such a case, we usually try to change the value of x. The point (3, 2) is not feasible. If you recall the feasible set, (3, 2) is somewhere outside.

Let us try giving a feasible point. Let us say I give (½, 0). Let us see what happens. The same error occurs; it says singular matrix again. Let us try changing the value of μ from (0, 0, 0) to (1, 1, 1).

Again, I am getting the same issue; it says I am having a singular matrix. Let us try a point much farther outside. I am going to try a point in the negative axis of x_1 and x_2, which is certainly not in the feasible set.

Let me give (-2, -1) and check what happens now. Of course, you can see that it says singular matrix again, but what you can see is that one step has passed; we have come to a different point.

At the first point (-2, -1), the matrix was not singular; the Hessian matrix was not singular. But at the next point, which is (1.5, 1), the matrix has actually become singular.

This tells us that for most points, it appears that the Hessian matrix is singular, but possibly there are certain points where it is not singular. That is certainly a problem because if I have to search for the right point everywhere, and even then we have not got the answer because I somehow found a point where the Hessian is non-singular, but when the trajectory moves to another point like (1.5, 1), the matrix becomes singular again. That is not good either because until I reach the answer, at every point I need to have the Hessian matrix be non-singular. Otherwise, I cannot proceed. So why is this happening? Is there a result which says that the Hessian matrix is singular in many places if the problem is a linear programming problem? We will discuss that.

Let us consider a general linear programming problem. Much earlier we gave the general structure. This is the general structure of a linear programming problem, where the objective function as well as the inequality constraints and the equality constraints are linear. I will write down this problem and we will discuss how it can be modified.

We have this particular problem where we are

minimizing $c^T x$ subject to $a_i^T x - b_i \leq 0$ for all i in 1 to p, and $\bar{a}_j^T x - \bar{b}_j = 0$ for all j equal to 1 to m.

If I put all these p constraints together, I can write them as $Ax - b \leq 0$, where A is a $p \times n$ matrix. The first row of A will have a_1^T, the second row will have a_2^T, and so on; the i-th row will have a_i^T. You have p rows in all. So A is a $p \times n$ matrix, x is an $n \times 1$ matrix, and b is a $p \times 1$ matrix with entries $b_1, b_2, ..., b_p$.

In a similar way, I can write down the equality constraints as $\bar{A}x - \bar{b} = 0$, where $\bar{A}$ is an $m \times n$ matrix, x is an $n \times 1$ matrix, and $\bar{b}$ is an $m \times 1$ matrix.

I have written all these constraints in the form of matrix inequalities and equalities. You can do this for any linear programming problem or any constraint set defined using linear constraints.

Now let us analyze what happens to the Hessian of the augmented Lagrangian function when we are working with a linear programming problem. We have the expression for the Hessian matrix. Note that since f is a linear objective function, its Hessian is 0.

The h_j's are linear constraints, so their Hessians are also 0. So those terms go to 0. However, the terms $\nabla h_j \nabla h_j^T$ remain. In a similar way, the g_i's are linear inequality constraints, so ∇g_i is a constant vector (not zero), but the Hessian of g_i is 0.

The term $\nabla g_i \nabla g_i^T$ need not be zero because it is just the outer product of the gradient. What this means is that the Hessian will only have the term $\sum \nabla h_j(x) \nabla h_j(x)^T$ and the term involving the indicator function for the inequality constraints: $\sum [g_i(x) + \mu_i/(2\gamma) \geq 0] \nabla g_i(x) \nabla g_i(x)^T$.

So for a linear programming problem, the Hessian of the augmented Lagrangian function is just $2\gamma \times [\sum_{j=1}^{m} \nabla h_j(x) \nabla h_j(x)^T + \sum_{i=1}^{p} [g_i(x) + \mu_i/(2\gamma) \geq 0] \nabla g_i(x) \nabla g_i(x)^T]$.

The Hessian becomes simplified this way.

Let us consider the case when you have only equality constraints in the linear program.

Case 1 is the problem min $c^T x$ subject to $\bar{A}x = \bar{b}$. In that case, you do not have the ∇g_i component, so that becomes 0. Now there is something we need to look at.

What is ∇h_1? ∇h_1 is just $\bar{a}_1$. The first constraint is $\bar{a}_1^T x = \bar{b}_1$, so if you take the gradient, it is just $\bar{a}_1$. For the second constraint, it is $\bar{a}_2$, for the third it is $\bar{a}_3$, and so on, for the m-th constraint it is $\bar{a}_m$. The whole summation $\sum \nabla h_j(x) \nabla h_j(x)^T$ can be written as $\bar{A}^T\bar{A}$.

The constant 2γ is outside. So it is just the product of the matrices $\bar{A}^T$ and $\bar{A}$. We are checking whether this matrix is invertible or not. Basic linear algebra tells you that $\bar{A}$ is an $m \times n$ matrix, so $\bar{A}^T\bar{A}$ is an $n \times n$ matrix.

When would this be non-singular? It would be non-singular if the rank of $\bar{A}^T\bar{A}$ is equal to n. This happens if and only if the rank of $\bar{A}$ is equal to n. If the rank of $\bar{A}$ is equal to n, then $\bar{A}^T\bar{A}$ is a non-singular matrix; its rank will be n.

But when would the rank of $\bar{A}$ be equal to n? The first thing is that m has to be greater than or equal to n. If $m < n$, then there is no way that the rank of $\bar{A}$ is going to be equal to n, because the rank is less than or equal to the number of rows and the number of columns. If $m < n$, then the rank of $\bar{A}$ is $\leq m$, which is strictly less than n.

So if the number of equality constraints is less than n, the rank of $\bar{A}$ is never going to be n. If $m \geq n$, we may have rank of $\bar{A} = n$. But there is a problem with that as well.

Why? Because suppose you have an $\bar{A}$ matrix with rank n. Then you know that the set of solutions for $\bar{A}x = b$ is a singleton set. So rank of $\bar{A} = n$ implies that the set of x for which $\bar{A}x = b$ is a single point. What does this tell you? For this particular problem, you either have rank of $\bar{A} < n$, in which case the Hessian $\bar{A}^T\bar{A}$ is singular and the method fails. Or, if we are fortunate and have rank of $\bar{A} = n$, then we do not have to struggle with the augmented Lagrangian method; we already have the answer because $\bar{A}x = b$ has only a unique solution.

That is the answer. If you have a unique solution, that is the answer; you do not need to process the problem any further. So what this tells us is that either the problem is trivial when rank of $\bar{A} = n$, or when rank of $\bar{A} < n$, the Hessian is singular, which means you cannot use this method. That is what happens when you have only equality constraints.

Now what happens when you have inequality constraints? This was the special case of only equality constraints. The next question is what happens when you have inequality constraints as well. Let us check that too.

Let me consider the case when you have only inequality constraints, which is $Ax \leq b$. In this case, you do not have the equality constraint, so the $\nabla h_j \nabla h_j^T$ term goes to 0.

You only have the $\nabla g_i \nabla g_i^T$ term, and ∇g_i will be the rows of A^T. Writing down the Hessian of L_γ is simple; it is just $2\gamma \times \sum_{i=1}^{p} [g_i(x) + \mu_i/(2\gamma) \geq 0] \nabla g_i(x) \nabla g_i(x)^T$.

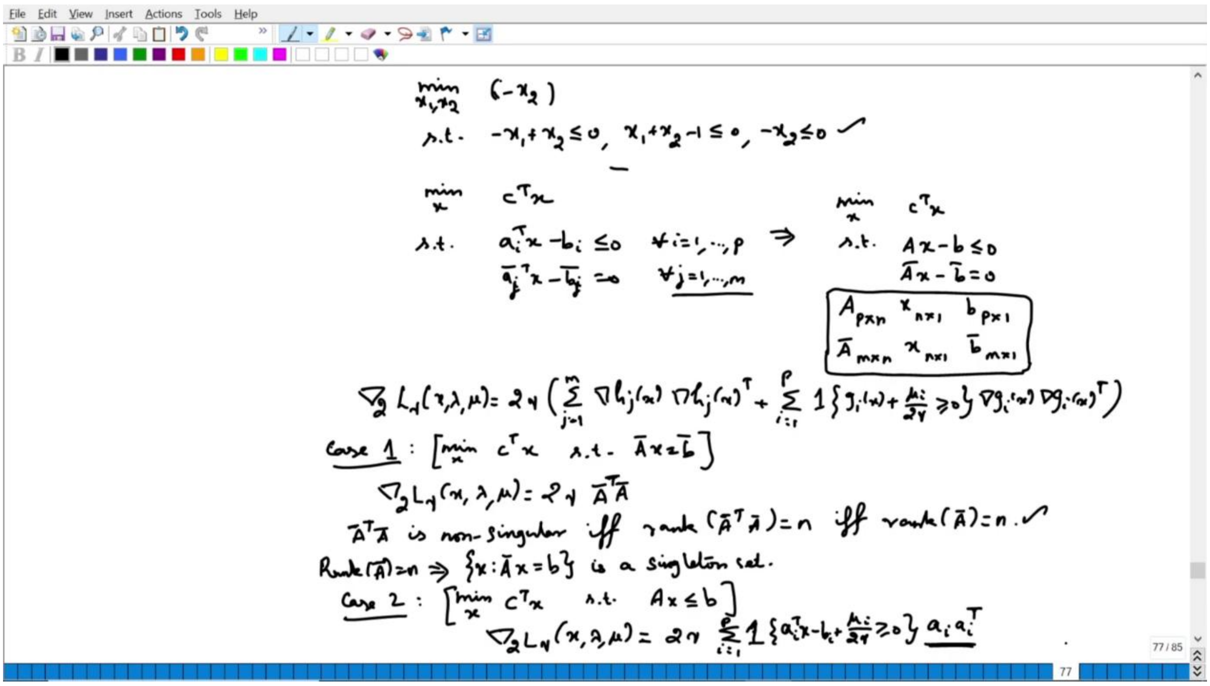
I will write this as $\sum_{i=1}^{p} [g_i(x) + \mu_i/(2\gamma) \geq 0]\, a_i a_i^T$, where a_i is a column vector (the gradient of the i-th constraint). The column vector times its transpose becomes a matrix.

So you have this matrix as the Hessian matrix. You might be confused about how to handle this indicator function.

What it says is simple: unless you satisfy the condition $g_i(x) + \mu_i/(2\gamma) \geq 0$, the term $a_i a_i^T$ is not included. Since $g_i(x)$ is $a_i^T x - b_i$, the condition is $a_i^T x - b_i + \mu_i/(2\gamma) \geq 0$.

If this condition holds for a constraint, the term is included; if not, it is excluded. It is like removing certain rows from A or certain columns from A^T. You can call this selected set B. The product of those matrices is what you get as the Hessian.

(Refer Slide Time 25:21)



Just as an example, consider the problem with only inequality constraints written as $Ax - b \leq 0$. For our triangle example, we have $a_1 = (-1, 1)$, $a_2 = (1, 1)$, $a_3 = (0, -1)$.

So the matrix

A is [[-1, 1], [1, 1], [0, -1]], and A^T is [[-1, 1, 0], [1, 1, -1]].

Now consider the point x = (3, 2) and μ = (0, 0, 0).

Let's check the Hessian matrix.

We have $\nabla^2 L\_\gamma(x, \lambda, \mu) = 2\gamma \times \sum [a_i^T x - b_i + \mu_i/(2\gamma) \geq 0]\, a_i a_i^T$ for i=1 to 3.

The b values are $b_1=0$, $b_2=1$, $b_3=0$. For i=1: $a_1^T x - b_1 = (-3 + 2) - 0 = -1$, which is < 0, so this term is 0.

For i=2: $a_2^T x - b_2 = (3+2) - 1 = 4$, which is ≥ 0, so we include $a_2 a_2^T$. a_2 is (1, 1), so $a_2 a_2^T$ is [[1, 1], [1, 1]]. For i=3: $a_3^T x - b_3 = (-2) - 0 = -2$, which is < 0, so this term is 0.

So the Hessian is 2γ * [[1, 1], [1, 1]], which is a singular matrix.

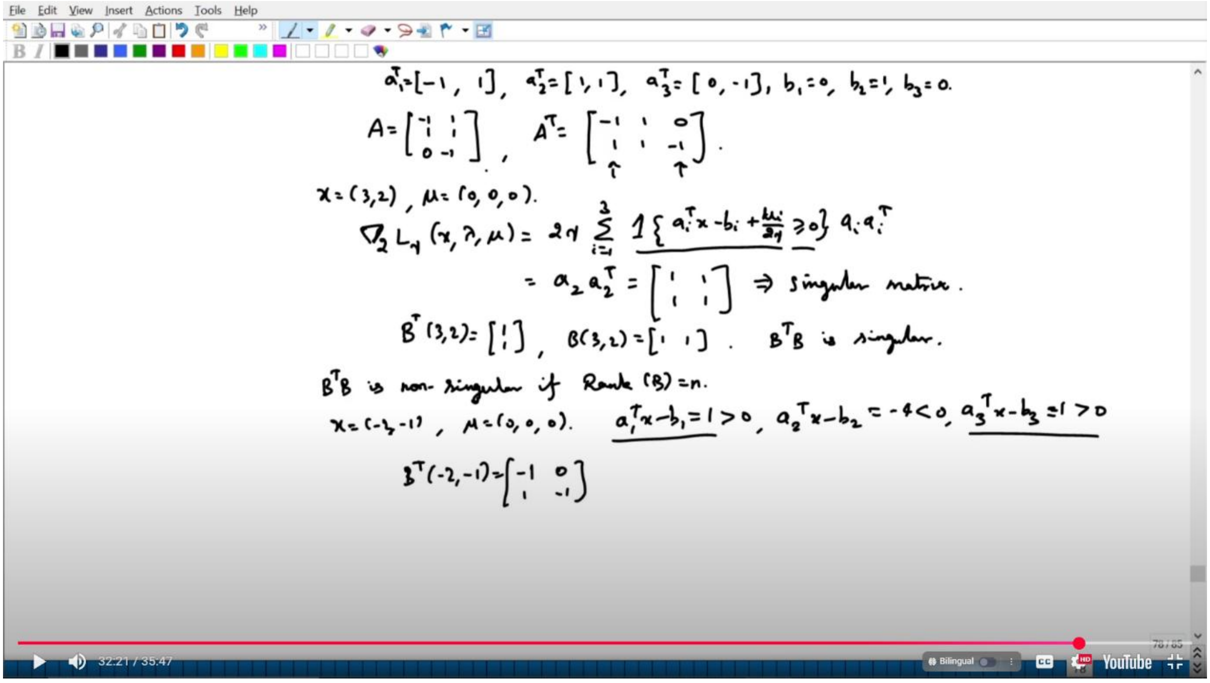
Even though the full matrix A has rank 2 (which is n), we got a singular matrix because for two of the constraints the condition was not met, so we had to remove them. The Hessian, when you have only inequality constraints, will be some subset of columns from A^T.

Let's call this subset B. In this case, for x=(3,2), B^T is just (1, 1), so B is (1, 1), and $B^T B$ is singular. $B^T B$ will be non-singular if the rank of B is n. This is possible.

For example, if you had given x=(-2, -1) and μ=(0,0,0), for the first constraint: $a_1^T x - b_1 = (2 - (-1))$? Let's calculate properly. $a_1^T x = (-1)*(-2) + (1)*(-1) = 2 - 1 = 1$. $1 - 0 = 1 > 0$. So include. For the second constraint: $a_2^T x - b_2 = (1)*(-2) + (1)*(-1) = -2 -1 = -3$. $-3 - 1 = -4 < 0$. So exclude. For the third constraint: $a_3^T x - b_3 = (0)*(-2) + (-1)*(-1) = 1$. $1 - 0 = 1 > 0$.

So include. So B^T would be the vectors for the first and third constraints: (-1, 1) and (0, -1). The rank of this is 2, so $B^T B$ was non-singular. So when we used (-2, -1), $B^T B$ was non-singular. When we used (3, 2), $B^T B$ was singular.

(Refer Slide Time 32:21)



So when you have inequality constraints, it turns out that even for non-trivial problems we can have a non-singular Hessian. But that has to happen for every single point that goes in the trajectory. We have a trajectory of points before we converge to the answer.

For every point in the trajectory, we need B^TB to be non-singular; only then will the method converge to the right answer. We can construct problems where that happens, but for most given problems you will see that these conditions are not satisfied. If even in between you have one point where the Hessian becomes singular, your problem cannot be solved using the augmented Lagrangian method.

The Hessian becomes singular, and that happens almost every time. When you have only equality constraints, it happens every time except for the trivial case. And for inequality constraints, you might have many points in the trajectory where the Hessian could be singular. That is the reason we avoid using the augmented Lagrangian method to solve linear programming problems.

Actually, it is the same for the quadratic penalty method as well. If you check the Hessian of the quadratic penalty method, you will see that the Hessian of f is 0, the Hessians of g_i are 0, and you will only have the $\nabla g_i \nabla g_i^T$ terms, just as we had for the augmented Lagrangian method. This is for inequality constraints. When you have both, the structure is similar.

I have written down the Hessian for equality constraints, and of course you will have that term alone; you will not have $\nabla^2 f$ and $\nabla^2 h_j$.

Similarly, for inequalities you will not have $\nabla^2 f$ and $\nabla^2 g_i$, but you will only have $\nabla g_i \nabla g_i^T$. I think that shows why we avoid using these two methods—the augmented Lagrangian method and the quadratic penalty method—for linear programming problems.

Of course, for non-linear programming problems, these methods work well, as we have seen in the examples. In the next week, we will start with linear programming problems and the methods related to linear programming problems. Thank you.