Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematics

Institute of IIT Kanpur

Lecture: 48

Hello everyone, this is the third lecture of week 10. Recall that in the previous lecture we were learning about the affine scaling method, which is one of the interior point methods. That means we start at an interior point and traverse through interior points before we reach the solution. The solution is possibly a vertex, but we traverse through the interior points. The idea that we saw in the previous lecture was that when you start at an interior point, we need to choose a direction in such a way that it is a descent direction as well as in such a way that it lies within the feasible set.

What we did was we just projected the steepest descent direction, -c, to the null space of A, and that is given by the mathematical expression: $(I - A^T(A A^T)^{-1}A)$ multiplied by -c, which is $- (I - A^T(A A^T)^{-1}A)$ c. That was one of the steps.

We actually have another step in the affine scaling method. Instead of projecting x, or x_k to be precise, what we do is we actually do something called centering. We center x_k with respect to the feasible set, do this projection, and then convert it back to the original form; that is de-centering. First, I will explain what centering means and then I will put the steps together after that.

Suppose we fix a vector. We usually fix this vector. I define the vector 1 as the vector of all ones. That means it is an n-length vector of all ones. This is found to be efficient, which is the reason it is actually done in the affine scaling method.

How do you center it?

You have x_k. Given a vector x_k, perform the transformation:

$y_k = X_k^{-1}x_k$

where capital X_k is the diagonal matrix consisting of the elements of x_k.

That is, it is a diagonal matrix with diagonal entries $[x_{k1}, x_{k2}, ..., x_{kn}]$.

x_k is a vector $[x_{k,1}, x_{k,2}, ..., x_{k,n}]$.

Note: $\mathbf{x}_{k,n}=x_k^n$, $X_k=X^k$ and $x_k=x^k$

If you construct a matrix with those entries as the diagonal entries and the others as 0, that is what is called a diagonal matrix. That is what is capital X_k.

When you do this, you can see that y_k is actually the one vector.

That is very simple because X_k^{-1} will be $[1/x_{k,1}, 1/x_{k,2}, ..., 1/x_{k,n}]$.

If you multiply that by $[x_{k,1}, x_{k,2}, ..., x_{k,n}]$, you will get [1, 1, ..., 1].

What we are actually going to do is, given any x_k, we first convert it into the one vector, then apply the steepest descent, and then convert it back to the x domain.

How do you convert it back to the x domain? Very simple, you have to pre-multiply on both sides by X_k. We also have capital $X_k\, y_k = x_k$.

So, if you want to center a particular vector, you multiply by capital X_k^{-1}, and if you want to de-center it, you multiply by X_k.

That is basically what the centering and de-centering process is.

Now putting all of them together, what we do is suppose you have a vector x_k.

The steps are as follows.

Suppose we have a vector x_k.

In the first step, what I am going to do is find:

$y_k = X_k^{-1}x_k$

which is actually just the one vector.

Now, we take the projected direction we calculated earlier:

$d_k = - (I - A^T(A\,A^T)^{-1}A)\ c$

You can see that none of this depends on k.

A, A^T, and c are all constants.

With this, you write the new centered point as:

y_new = 1 + α

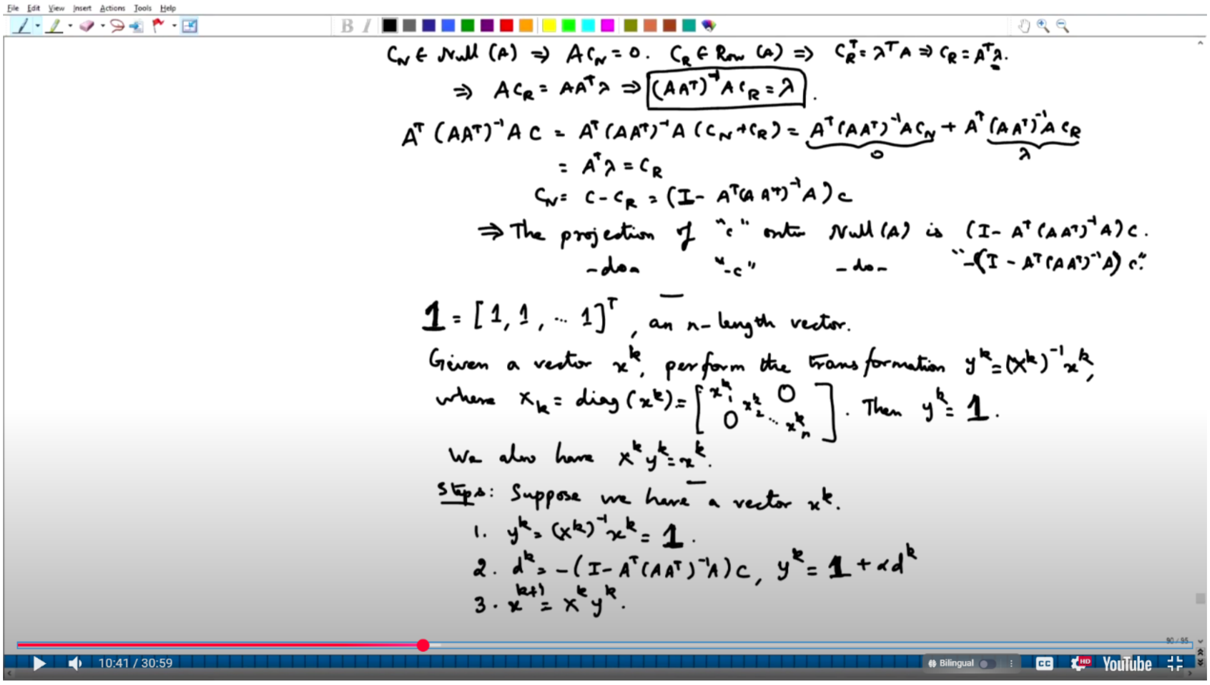
We will see what this α is a little later, but nevertheless, for some α, the new point will be 1 + α d_k.

Now you have to convert this new point back to the x domain.

So, x_{k+1} is actually capital X_k times the new y point:

$x_{k+1} = X_k$ (y_new) $= X_k (1 + \alpha d_k)$

(Refer Slide Time 10:30)



$c_N \in \text{Null}(A) \Rightarrow Ac_N = 0$. $c_R \in \text{Row}(A) \Rightarrow c_R^T = \lambda^T A \Rightarrow c_R = A^T\lambda$.

$\Rightarrow Ac_R = AA^T\lambda \Rightarrow \boxed{(AA^T)^{-1}Ac_R = \lambda}$.

$A^T(AA^T)^{-1}Ac = A^T(AA^T)^{-1}A(c_N + c_R) = \underbrace{A^T(AA^T)^{-1}Ac_N}_{0} + A^T\underbrace{(AA^T)^{-1}Ac_R}_{\lambda}$

$= A^T\lambda = c_R$

$c_N = c - c_R = (I - A^T(AA^T)^{-1}A)c$

$\Rightarrow$ The projection of "c" onto Null (A) is $(I - A^T(AA^T)^{-1}A)c$.

-do- "$-c$" -do- "$-(I - A^T(AA^T)^{-1}A)c$"

$\bar{\mathbf{1}} = [1, 1, \cdots 1]^T$, an n-length vector.

Given a vector x^k, perform the transformation $y^k = (X^k)^{-1}x^k$,

where $X_k = \text{diag}(x^k) = \begin{bmatrix} x_1^k & & 0 \\ & x_2^k & \\ 0 & & \ddots \; x_n^k \end{bmatrix}$. Then $y^k = \mathbf{1}$.

We also have $X^k y^k = x^k$.

<u>Steps</u>: Suppose we have a vector x^k.

1. $y^k = (X^k)^{-1}x^k = \mathbf{1}$.
2. $d^k = -(I - A^T(AA^T)^{-1}A)c$, $y^k = \mathbf{1} + \alpha d^k$
3. $x^{k+1} = X^k y^k$.

If you want to explain this in words, you start with an initial point, let us say x_0. How do you go to x_1?

1. You first center it to 1 (by calculating $y_0 = X_0^{-1}x_0$).
2. Choose the direction which is $d_k = -(I - A^T(A A^T)^{-1}A)$ c.
3. Add it to the centered vector 1 with a step size α.
4. De-center it by multiplying by capital X_0.

By which you actually get x_1.

You started with x_0, you are getting x_1 by this process. I should write it as x_{k+1}.

So, there is actually a catch because we have got the steps. To an extent, it is similar to what we did for gradient descent, just that there are some projection steps, centering and de-centering steps. But the biggest problem that we have here is that we do not have a stopping criterion.

So, there is actually a catch because we have got the steps. To an extent, it is similar to what we did for gradient descent, just that there are some projection steps, centering and de-centering steps. But the biggest problem that we have here is that we do not have a stopping criterion.

If you recall what you want for completing an optimization algorithm, we want to choose a direction, we want to choose a step size, and at the end we should also have a stopping criterion. What is the stopping criterion here? We can keep moving around, but when can we decide that we have actually converged to the solution? For this, we actually need the basics of duality.

As far as optimization is concerned, if you have done a course on optimization, maybe you have also learnt about duality, but I will give a quick recap. I am not going to do it in detail because that is not the purpose of this course, but in case you are not well versed with duality, I will do it quickly. We will stop affine scaling and we will move to duality; we will come back to affine scaling after this.

Consider a general constrained optimization problem. You have this form:

min f(x)

subject to

$g_i(x) \leq 0$ for all i in 1 to p

and

$h_j(x) = 0$ for all j in 1 to m.

While solving this, we first form the Lagrangian which is:

$L(x, \lambda, \mu) = f(x) + \sum \lambda_i\, g_i(x)$ for $i = 1$ to $p + \sum \mu_j\, h_j(x)$ for $j = 1$ to m

and let us say that $\lambda_i \geq 0$ for all i. This is the Lagrangian.

What I am claiming is that this particular problem is actually equivalent to:

min over x max over $\lambda \geq 0$ and μ (where $\mu \in \mathbb{R}^m$ taking any possible values) of $L(x, \lambda, \mu)$.

This duality, or rather any given optimization problem, can be written as a min-max problem, where the minimization is over x and the maximization is over all λ and μ, where λ is non-negative and μ is unconstrained.

Why is that? Suppose you take some x that is not feasible. That means the feasible set here is x such that $g_i(x) \leq 0$ for all i and $h_j(x) = 0$ for all j.

If x does not belong to F, that means at least one of the p + m constraints does not hold.

* Say $g_i(x) > 0$, for example. Then, max over $\lambda \geq 0$, $\mu \in \mathbb{R}^m$ of $L(x, \lambda, \mu)$ is ∞.

Why is that? Because λ_i is available only as a coefficient multiplying $g_i(x)$. If $g_i(x)$ were positive, I can drive λ_i as high as I want and thus we will actually have the maximum being unbounded. If it fails any of the inequality constraints, then you can see that the max is ∞.

* What happens if $h_j(x) \neq 0$? Then again, the max of this quantity is ∞. μ_j is available only as a coefficient to $h_j(x)$.

So if $h_j(x)$ is positive, I would drive μ_j as high as ∞, and thus the maximum will be ∞.

If h_j is negative, then I will drive μ_j as low as $-\infty$, so the maximum of L will again be ∞.

This tells us that this quantity is ∞ if x is not a part of the feasible set.

What is it if it is a part of the feasible set? Suppose x satisfies all inequality constraints and all equality constraints. In that case, I claim that if $x \in F$, then max over $\lambda \geq 0$, $\mu \in \mathbb{R}^m$ $L(x, \lambda, \mu)$ is actually just f(x).

Why is that?

* If $x \in F$, we have $h_j(x) = 0$. That means $\sum \mu_j h_j(x) = 0$ for any μ_j.

* Similarly, if $g_i(x) \leq 0$ for all i, then for all $\lambda \geq 0$, you will see that $\sum \lambda_i g_i(x) \leq 0$.

* So if you want to maximize, you can choose all λ's to be equal to 0. That puts it at the maximum value, which is 0.

That actually tells us that when $x \in F$, you have the max of the Lagrangian, if you maximize the Lagrangian over all λ non-negative and μ unconstrained, you will have the maximum to be f(x).

This tells us that:

min over x max over $\lambda \geq 0$, μ $L(x, \lambda, \mu) = f(x)$ if x is in the feasible set, and ∞ if x is not in the feasible set, which is equivalent to saying that you are minimizing f(x) subject to x belonging to the feasible set.

I have not started duality as yet.

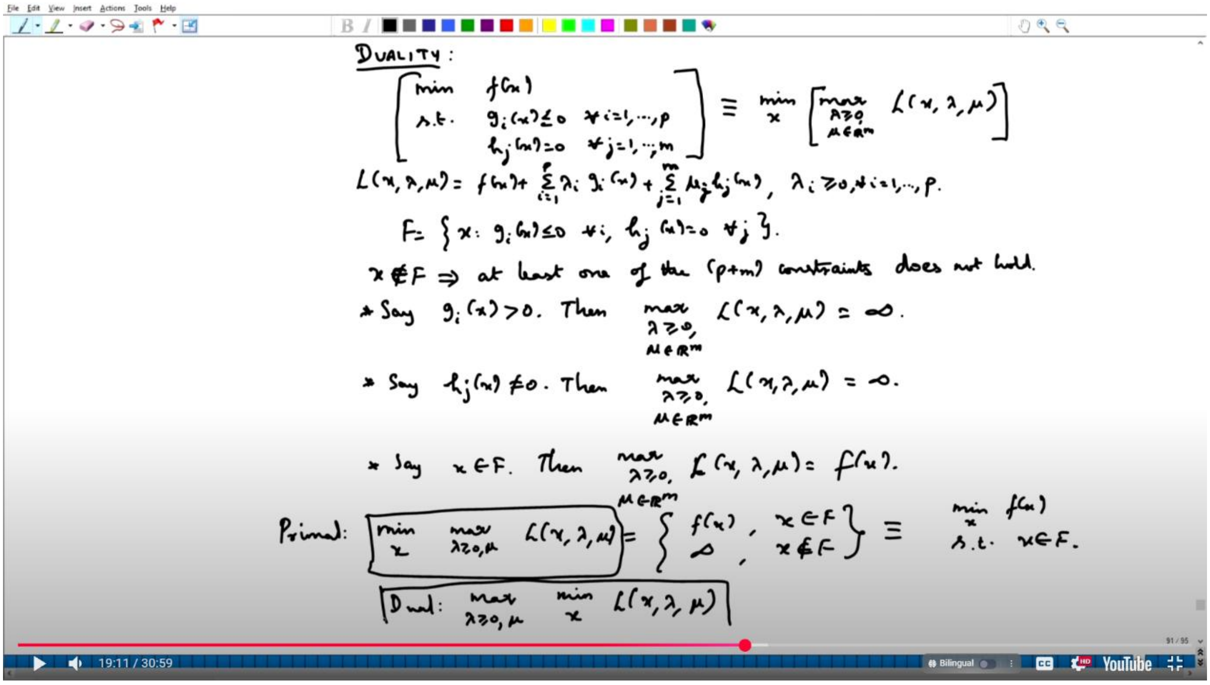
What I have just said is that we can rewrite any given constrained optimization problem as a min-max problem, min over x, max over λ, μ, where λ is constrained to be non-negative and μ is not constrained at all. This is called the primal problem.

Now, the **dual problem** is very easy to represent. You just interchange the min and max.

So, it is max over $\lambda \geq 0$, μ min over x $L(x, \lambda, \mu)$. This is the primal problem. The dual problem is max over $\lambda \geq 0$, μ min over x $L(x, \lambda, \mu)$.

This is another optimization problem; both of them need not be the same.

(Refer Slide Time 19:17)



DUALITY:

$$\begin{bmatrix} \min & f(x) & \\ \text{s.t.} & g_i(x) \leq 0 & \forall i = 1, \dots, p \\ & h_j(x) = 0 & \forall j = 1, \dots, m \end{bmatrix} \equiv \min_{x} \left[\max_{\substack{\lambda \geq 0 \\ \mu \in \mathbb{R}^m}} L(x, \lambda, \mu) \right]$$

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) + \sum_{j=1}^{m} \mu_j h_j(x), \quad \lambda_i \geq 0, \forall i = 1, \dots, p.$$

$$F = \{ x : g_i(x) \leq 0 \ \forall i, \ h_j(x) = 0 \ \forall j \}.$$

$x \notin F \Rightarrow$ at least one of the $(p+m)$ constraints does not hold.

* Say $g_i(x) > 0$. Then $\max\limits_{\substack{\lambda \geq 0, \\ \mu \in \mathbb{R}^m}} L(x, \lambda, \mu) = \infty$.

* Say $h_j(x) \neq 0$. Then $\max\limits_{\substack{\lambda \geq 0, \\ \mu \in \mathbb{R}^m}} L(x, \lambda, \mu) = \infty$.

* Say $x \in F$. Then $\max\limits_{\substack{\lambda \geq 0, \\ \mu \in \mathbb{R}^m}} L(x, \lambda, \mu) = f(x)$.

Primal: $\boxed{\min\limits_{x} \max\limits_{\lambda \geq 0, \mu} L(x, \lambda, \mu)} = \begin{cases} f(x), & x \in F \\ \infty, & x \notin F \end{cases} \equiv \begin{array}{l} \min\limits_{x} f(x) \\ \text{s.t. } x \in F. \end{array}$

$\boxed{\text{Dual: } \max\limits_{\lambda \geq 0, \mu} \min\limits_{x} L(x, \lambda, \mu)}$

The objective function of the primal problem and the objective function of the dual problem are they going to be the same? They need not be so. They are the same only when some constraints hold.

For example, if it is a convex optimization problem with some constraint qualifications holding. If the objective function is convex and the feasible set is also convex, then yes, with some constraint qualifications holding, the primal and the dual will have the same objective function value; the value of the primal and the value of the dual will be the same.

I am not going into those theories, but something that I want you to know is that the objective function value of the primal problem and the objective function value of the dual problem will be the same if the constrained optimization problem is a linear programming problem. Of course, there are some more constraints as well; it must not be unbounded or the feasible set must not be empty.

These are pathological cases or degenerate cases. If you avoid the degenerate cases, what I am trying to say is that when we consider a linear programming problem where the solution exists, that necessarily means that it is not unbounded (if it is unbounded then no solution exists) and it also means that the feasible set is non-empty (if the feasible set is empty again the solution does not exist). Then the values of the objective functions of the primal and the dual are one and the same.

We are not going to prove this, but this is to actually tell you that we are going to use this for obtaining a stopping criterion, because we are considering a linear programming problem. If it happens to be a problem where it is not unbounded and the feasible set is non-empty, then I can use this result.

First, we will actually try to find what is the dual of the LP under consideration. The primal problem that we have is:

minimize $c^T x$

subject to

$A x = b$,

$x \geq 0$.

We will write down the Lagrangian $L(x, \lambda, \mu)$. It is going to be:

$L(x, \lambda, \mu) = c^T x + \mu^T(A x - b) - \lambda^T x$.

These are equality constraints; for equality constraints we use μ. For the inequality constraints $x \geq 0$, we write them as $-x \leq 0$, so we use $-\lambda^T x$, with $\lambda \geq 0$.

The primal problem can be written as:

min over x max over $\lambda \geq 0$, $\mu \in \mathbb{R}^m$ $L(x, \lambda, \mu)$.

The dual is:

max over $\lambda \geq 0$, $\mu \in \mathbb{R}^m$, min over x $L(x, \lambda, \mu)$.

We will first solve this problem: we are fixing λ and μ and going to solve min over x $L(x, \lambda, \mu)$. So, we are going to solve:

min over x of $c^T x + \mu^T(A x - b) - \lambda^T x$.

To minimize this, look at the derivative of the objective function with respect to x. The gradient is:

$c + A^T\mu - \lambda$.

If this gradient is not zero, then the minimum is $-\infty$ because the expression is linear in x.

So, for the minimum to be finite, we must have:

$c + A^T\mu - \lambda = 0$,

or

$A^T\mu - \lambda = -c$,

which is equivalent to

$c + A^T\mu = \lambda$.

If this holds, then the objective function becomes $-\mu^T b$ (because the terms involving x cancel out: $c^T x + \mu^T A x - \lambda^T x = 0$ since $\lambda = c + A^T\mu$).

So, the minimized value is $-\mu^T b$.

If $c + A^T\mu \neq \lambda$, then the minimum is $-\infty$. Therefore, to maximize the dual function, we must choose μ and λ such that $c + A^T\mu = \lambda$.

Then the dual becomes:

max over $\lambda \geq 0$, μ $-\mu^Tb$

such that

$c + A^T\mu = \lambda$ and $\lambda \geq 0$.

Since $\lambda = c + A^T\mu$, the constraint $\lambda \geq 0$ becomes $c + A^T\mu \geq 0$, or $A^T\mu \geq -c$. We can remove λ because it is determined by μ.

So the dual problem is:

max over μ $-\mu^Tb$

subject to

$A^T\mu \geq -c$.

This is equivalent to:

min over μ μ^Tb

subject to

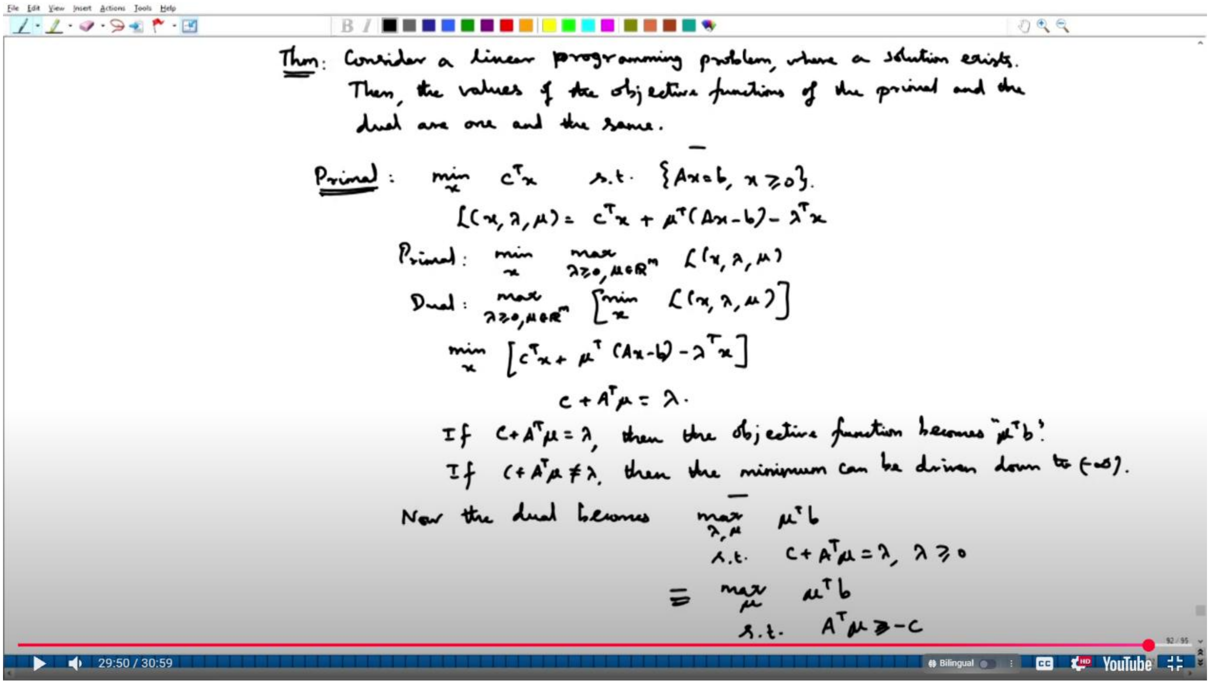
$A^T\mu \geq -c$.

This turns out to be the dual problem. The reason we struggled to derive the dual of the particular problem is to find a stopping criterion.

The stopping criterion is that $c^Tx - (-\mu^Tb) = c^Tx + \mu^Tb$ should be as close to 0 as it can be.

You will perform these steps until $c^Tx + \mu^Tb$ is less than or equal to some tolerance. In that case, we actually stop. That is going to be the stopping criterion.

(Refer Slide Time 30:31)



<u>Thm</u>: Consider a linear programming problem, where a solution exists. Then, the values of the objective functions of the primal and the dual are one and the same.

<u>Primal</u>: $\min_{x} \; c^T x \quad$ s.t. $\{Ax = b, \; x \geqslant 0\}$.

$$L(x, \lambda, \mu) = c^T x + \mu^T (Ax - b) - \lambda^T x$$

Primal: $\min_{x} \max_{\lambda \geq 0, \mu \in \mathbb{R}^m} L(x, \lambda, \mu)$

Dual: $\max_{\lambda \geq 0, \mu \in \mathbb{R}^m} \left[\min_{x} L(x, \lambda, \mu) \right]$

$$\min_{x} \left[c^T x + \mu^T (Ax - b) - \lambda^T x \right]$$

$$c + A^T \mu = \lambda.$$

If $c + A^T \mu = \lambda$, then the objective function becomes "$\mu^T b$".

If $c + A^T \mu \neq \lambda$, then the minimum can be driven down to $(-\infty)$.

Now the dual becomes $\max_{\lambda, \mu} \; \mu^T b$

s.t. $c + A^T \mu = \lambda, \; \lambda \geqslant 0$

$$\equiv \max_{\mu} \; \mu^T b$$

s.t. $A^T \mu \geqslant -c$

We will continue in the next lecture. Thank you.