Lecture 17

The purpose of this lecture is to provide a brief introduction to the theory of duality in convex optimization. A very nice reference on this topic is the book *Convex Optimization* by Boyd & Vandenberghe, Cambridge University Press.

This lecture looks at systematic ways of minimizing real valued functions that may depend upon multiple variables subject to appropriate constraints. We shall exclusively work with *convex functions* and mostly with linear functions, that have the property that a local minimum is also a global minimum.

Recall that min cost multicast with network coding was a linear program (LP). An LP in standard form can be expressed as

$$\min \; c^T x \tag{1}$$

$$\text{subject to } Ax \leq b \tag{2}$$

where $A_{m\times n}$ and b and c are known vectors of length m and n respectively, and we are interested in finding an optimal $\boldsymbol{x}^*$. We observed previously that there exist efficient techniques for solving problems of this type. We now explain the concept of the dual of this linear program, through an example. Consider the following LP

$$\min -4x_1 - 5x_2 \tag{3}$$

$$\text{subject to } \begin{pmatrix} -1 & 0 \\ 2 & 1 \\ 0 & -1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 3 \\ 0 \\ 2 \end{pmatrix} \tag{4}$$

On inspection we realize that the first and the third constraints enforce positivity of the variables x_1 and x_2. Now, suppose that we tried using the constraints to get a useful lower bound on the value of the objective. For example multiplying the fourth constraint by two yields $2x_1 + 4x_2 \leq 6$. We already know that $2x_1 + x_2 \leq 3$ i.e. we can conclude that $4x_1 + 5x_2 \leq 9$ i.e. $-4x_1 - 5x_2 \geq -9$.

The question we ask is, can this lower bound be achieved i.e. does there exist a value of the pair (x_1^*, x_2^*) such that it satisfies all the inequalities and has an objective of value -9. Achieving the lower bound would imply minimization is complete. On inspection we realize that $x_1^* = 1$, $x_2^* = 1$ achieves the lower bound.

An interesting question to ask is whether we got lucky this time, or whether achievable lower bounds can be realized by systematically considering linear combinations of the lower bound inequalities. It turns out that this process can indeed be done systematically if *strong duality* holds in the optimization problem. In the case of a linear program, the feasibility of the linear program is sufficient for strong duality to hold.

In the general case we proceed as follows.

Corresponding to the standard form LP we define the Lagrangian

$$L(x, \lambda) = C^T x + \sum_{i=1}^{n} \lambda_i (a_i^T x - b_i) \tag{5}$$

where a_i^T is the i^{th} row of A and $\lambda_i \geq 0$, $i = 1, \ldots, n$.

If a vector x_1 is feasible i.e. $Ax_1 \leq b$ then clearly $c^T x_1 + \sum \lambda_i (a_i^T x_1 - b_i) \leq c^T x_1$ since $\lambda_i \geq 0$ and $a_i^T x_1 - b_i \leq 0$ for all i. Therefore we have in general,

$$L(x, \lambda) \leq c^T x \tag{6}$$

$$\text{Furthermore, } L(x, \lambda) \geq g(\lambda) = \min_{\tilde{x} \in \Re^n} L(\tilde{x}, \lambda) \tag{7}$$

$$\implies L(x, \lambda) \geq g(\lambda). \tag{8}$$

In particular if x^* is the minimizer of the LP, then $c^T x^* \geq L(x^*, \lambda) \geq g(\lambda)$. Note that $c^T x^*$ only depends on the optimal x^* and $g(\lambda)$ only depends upon another set of variables λ that only have to be positive. Clearly maximizing $g(\lambda)$ over $\lambda \succeq 0$ ($\succeq$ denotes component-wise inequality) yields the best lower bound on $c^T x^*$.

The interesting fact is that in fact if λ^* denotes the optimal value that maximizes $g(\lambda)$ subject to the positivity constraint and if strong duality holds in the optimization problem, then in fact $c^T x^* = p^* = d^* = g(\lambda^*)$. In the language of optimization theory it says that the optimal values of the primal and dual problem are the same. It should be noted that while strong duality is always holds in linear programs, in non-linear programs it sometimes does not hold and only weak duality holds, i.e. $p^* = C^T \geq g(\lambda^*) = d^*$.

Let us simplify the dual problem in our case to understand its structure. We observe that for some values of λ the bound produced by $g(\lambda)$ is useless. Note that

$$g(\lambda) = \min_{x \in \Re^n} L(x, \lambda) = \min_{x \in \Re^n} ((c^T + \lambda^T A)x - \lambda^T b) = \begin{cases} -\infty & C^T + \lambda^T A \neq 0 \\ -\lambda^T b & if C^T + \lambda^T A = 0 \end{cases} \tag{9}$$

(10)

i.e. for some values of λ, the value of the lower bound is actually $-\infty$ which is useless. Thus, the dual LP corresponding to $\max g(\lambda)$, s.t. $\lambda \succeq 0$ can be written as

$$\max - b^T \lambda \tag{11}$$

$$\text{subject to } C^T + \lambda^T A = 0 \tag{12}$$

$$\lambda \geq 0 \tag{13}$$

It turns out that examining the structure of the dual optimization often allows us to design more efficient and occasionally distributed solutions to the primal problem. We illustrate the possibility of designing distributed solutions by means of an example.

Consider a network with one source and one terminal where we want to support a flow of R. Suppose that we want to find the minimum cost flow (recall the discussion in previous lectures) that supports a rate of R over this network. For a flow of x_j over edge e_j the cost incurred is given by $\phi_j(x_j)$ where ϕ_j is a convex function. The constrained optimization problem can be posed as

$$\min \sum_{j=1}^{n} \phi_j(x_j) \tag{14}$$

$$\text{subject to } Ax = s, \tag{15}$$

where A is a the node-arc incidence matrix of the graph. More generally one would have capacity constraints on each edge but we choose to model them by assigning infinite cost to the solution if the capacity constraints are violated i.e. $\phi_j(x_j) = \infty$ if $x_j > C_j$ or $x_j < 0$.

As before the vector $s = \begin{cases} R & \text{at the source} \\ -R & \text{at the terminal} \\ 0 & \text{otherwise} \end{cases}$

Now consider the dual of this optimization problem. It is not an LP but the process of finding the dual remains the same. Note that since we have equality constraints, the dual variables need not necessarily be positive. We denote the dual variables by ν. We have,

$$L(x, \nu) = \sum \phi_j(x_j) + \nu^T(Ax - s) \tag{16}$$

$$= \sum_{j=1}^{n} (\phi_j(x_j) + \nu^T a_j x_j) - \nu^T s \tag{17}$$

where a_j is the j^{th} column of A corresponding to the j^{th} edge in the network.

To obtain the dual function we need to minimize $L(x, \nu)$ over x to obtain $g(\nu)$. Note that by definition, a_j has the entry $+1$ at the tail of the edge, the entry -1 at the head of the edge and zero elsewhere. Therefore if the j^{th} edge is denoted $(k \rightarrow l)$, then $\nu^T a_j x_j = (\nu_k - \nu_l)x_j$. i.e. if the values of ν_k and ν_l at nodes k and l are known then the optimization problem $\min_x L(x, \nu)$ decomposes into a set of n independent minimizations, one for each edge. The important point is that each edge can perform its own optimization as long as it knows the values of ν at its end nodes.

An algorithm for the distributed computation of the flows is now readily available. For a given ν the value of $g(\nu)$ can be computed. It remains to maximize the unconstrained function $g(\nu)$. It can be shown that the dual function is always concave (see the referenced text), therefore techniques such as subgradient optimization (which is similar to gradient descent) work. We basically start with an initial guess of the dual variable $\nu^{(0)}$ and iteratively keep refining it based on the current computation of the gradient.