MTH 995-001: Intro to CS and Big Data Spring 2015

Lecture 19 — March 26, 2015

Inst. Aditya Viswanathan *Scribe: Kenji Aono*

1 Overview

1.1 Last time

We were left with a couple of questions:

- The unboundedness of the dual of the standard form linear program (LP),
- Choice of the dual variables, λ and ν.

The issue seems to revolve around our choice of x, "what is x", which answers why the dual of the LP is unbounded from below unless it satisfies certain conditions. We should not confuse x being in the domain of the problem with x being feasible. From our definition of the standard form of the convex problem last lecture, x is only the set of values for which the function is defined, this does not mean that the values satisfy the constraints we defined earlier.

Example Given logarithms in the objective or constrained functions, then the set of all x which satisfy the constraint (a logarithmic in this case) will be the set of all x values which make the argument of the logarithmic positive. So we want something that is strictly positive. On the other hand, if we just have an affine function, the set of all possible x is all $\mathbb{R}^n$.

A subset of all possible values in the domain is going to be the set of all values which satisfy the constraints of the problem i.e. the set of all feasible x. The Lagrangian and the dual are just defined for all x in the domain and not only for feasible x. Which is why, when we wrote out the dual of the LP we could choose any value of x to make it unbounded from below. It is also the reason why we cannot choose λ, $\nu := 0$. In particular, by doing so we end up solving the unconstrained version of the problem. Think of the Lagrangian as a weighted combination of the constraints and the objective function: we use it to take a constrained formulation and form an unconstrained and since the goal is to minimize our objective function, we can simply minimize the Lagrangian. If we don't satisfy the constraints, the Lagrangian will grow very large; if it does satisfy the constraints, then each of those terms are such that they will be zero or less than zero and thus resulting in something small. **The moral is that** we have x in the domain, x being feasible, and among all the x being feasible, there is an optimal value.

1.2 Today

With respect to the extensions of the generalized inequalities we left off at last lecture – we may return to it later. This lecture will introduce a family of algorithms for solving an optimization

problem, in particular an inequality-constrained optimization problem. We will look at the **Barrier Method**, which is a subclass of the **Interior Point Method**. We will start with the scalar case (in the sense that the inequalities are scalar) and later extend it to general inequalities (disclaimer, we didn't get this far). Covering the entire basis for the algorithm goes beyond the scope of one lecture, hence we will assume that there is previous knowledge of the hierarchy of algorithms in the sense that we are aware of **Gradient Descent** or some descent-method involved in solving an unconstrained optimization problem. For an unconstrained-inequality problem, one can simply use one of the steepest-descent or similar methods. In the Equality-constrained minimization case, use the **Newton Method** (not the only method, but is a popular one). Then we are left with the inequality-constrained problems, which we will see can be solved using a series of equality-constrained.

We will also cover some material on **Generalized Inequality** that we missed last time (such as $x \succeq_k y \Leftrightarrow y - x \in K$), but this will be added to the previous lecture notes by *Sami Merhi*.

2 Inequality-Constrained

2.1 Interior Point Method

Consider a general optimization problem of the form

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 0,\ i \in (1, m), \\ &&& Ax = b. \end{aligned} \tag{1}$$

Where, $f_0, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are convex, twice continuously differentiable functions, and $A \in \mathbb{R}^{p \times n}$, $\operatorname{rank} A = p < n$. We will assume this is a linear inequality, and an optimal solution, x^*, exists with optimal value of p^*. Also assume that the problem is strictly feasible, that is, Slater's condition from the previous lecture holds – **strong duality exists**. The restriction on $\operatorname{rank} A$ is due to the overdetermined system having a different implementation (and a solution in the least-square sense), the following discussion is for "fat" matrices, where the number of columns is greater than the number of rows. It is typical that one would consider the constraints to be less than the dimensionality of the problem.

2.2 KKT Conditions

The optimality condition can be described using the Karush-Kuhn-Tucker conditions (see previous lecture for notes on KKT conditions) in the following form, where x^* is primal optimal and (λ^*, ν^*) are dual optimal:

$$\begin{aligned} Ax^* &= b, \\ f_i(x^*) &\leq 0,\ i = 1, \ldots, m, \\ \lambda^* &\geq 0, \\ \nabla f_o(x^*) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x^*) + A^\intercal \nu^* &= 0, \\ \lambda_i^* f_i(x^*) &= 0,\ i = 1, \ldots, m. \end{aligned} \tag{2}$$

The idea is to convert the inequality constraints in (2) to some sort of equality-constrained problem. Then we can solve the original problem as a linear problem, using the well-studied Newton method. With the original inequality-constrained, we are limited due to the iterative nature of the methods used to solve the problem, and we would like to avoid this.

2.2.1 Logarithmic Barrier

Define an indicator function and rewrite (1) as

$$\begin{aligned} &\text{minimize} && f_0(x) + \sum_{i=1}^{m} I_{-}(f_i(x)) \\ &\text{subject to} && Ax = b \end{aligned} \tag{3}$$

Where $I_{-} : \mathbb{R} \to \mathbb{R}$ is an indicator function

$$I_{-}(u) = \begin{cases} 0, & u \leq 0 \\ \infty, & u > 0. \end{cases}$$

If the inequality constraints are satisfied, the indicator function takes a value of zero, else it is infinity. It is, in effect, ensuring that our constraints are satisfied; for example, when we try to minimize in (3), we will already be at infinity cost when the constraints aren't met. If they are, the problem reduces to minimizing the objective function $f_0(x)$. So we now have (1)≡(3), but are still left with a problem that is not twice-differentiable. A solution to this is to consider an approximation to the indicator function of the form:

$$\hat{I}_{-}(u) = 0 \left(\frac{1}{t}\right) \log(-u)$$
$$\operatorname{dom} \hat{I}_{-} = -\mathbb{R}_{++}.$$

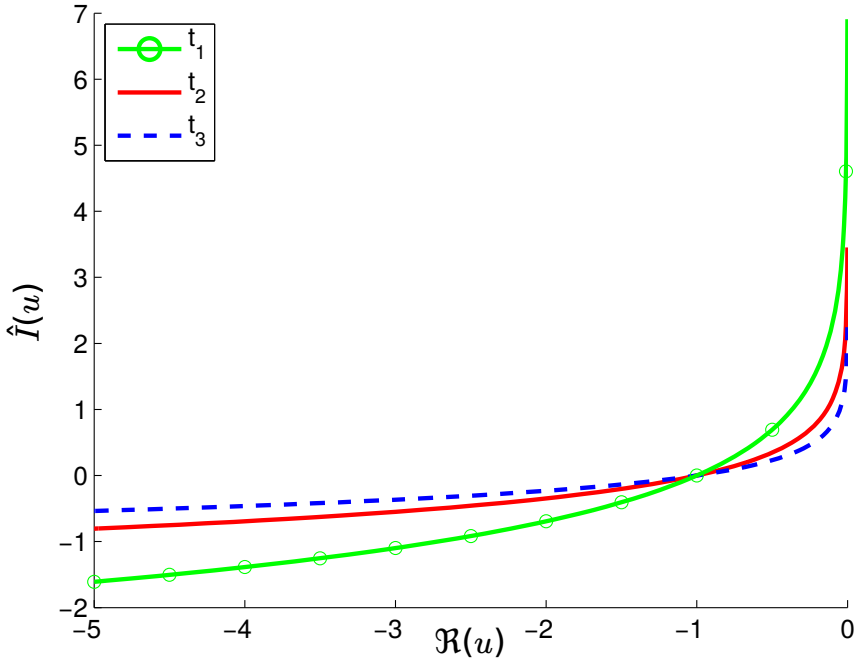

Figure 1: An approximation to the indicator function to enable differentiability.

This allows us to overcome the limitation that I_- is non-differentiable. Note that in Fig. 1, $t > 0$ is an accuracy parameter, and $t_1 < t_2 < t_3$. As t increases, we have a better approximation that is differentiable; however, for large t the Hessian has rapid oscillations and fluctuations and becomes **difficult to solve**. Thus, it is common to start with a moderate value of t, and gradually and sequentially increase the value and improve the approximation.

Definition 1 (Logarithmic Barrier)**.** *The function*

$$\Phi(x) = -\sum_{i=1}^{m} \log\left[-f_i(x)\right], \text{ with}$$
$$\operatorname{dom}\Phi = \left\{x \in \mathbb{R}^n \mid f_i(x) < 0,\ i = 1, \ldots, m\right\},$$

is called the log barrier of the problem in (1).

Note

$$\nabla\Phi(x) = \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla f_i(x)$$

$$\nabla^2\Phi(x) = \sum_{i=1}^{m} \frac{1}{f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^\intercal + \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla^2 f_i(x)$$

So if the function is twice-differentiable, the logarithm will also be twice-differentiable. We must start with a feasible point (the variant that does not need to meet this technicality is omitted from this discussion), then we can guarantee that we will stay in the feasible set. Also, since the formulation of the problem is such that we minimize the objective function, and $\Phi(x)$ increases as we approach the boundary of the feasible set, we also note that we will always tend towards the optimal value.

Rewrite (3) as follows:

$$\begin{array}{ll} \text{minimize} & tf_0(x) + \Phi(x) \\ \text{subject to} & Ax = b. \end{array} \tag{4}$$

Assume that (4) can be solved using Newton's method and that there is a unique solution for each $t > 0$, **denoted by** $x^*(t)$. This is not an unreasonable assumption if we recall that Slater's condition holds for the problem.

- Each $x^*(t)$ is called a **central point**.
- The entire sequence (set of points) of $x^*(t)$, $t > 0$ associated with (1) is the **central path**.

Points on the central path are characterized by the following necessary and sufficient conditions (assuming that Slater's condition holds):

$$x^*(t) \text{ is strictly feasible,}$$
$$Ax^*(t) = b,\ f_i(x^*(t)) < 0,\ i \in (1, m),$$

and by applying KKT conditions on (4), there exists a $\hat{\nu} \in \mathbb{R}^p$, such that

$$\begin{aligned} 0 &= t\nabla f_0(x^*(t)) + \nabla\Phi(x^*(t)) + A^\intercal\hat{\nu} \\ &= t\nabla f_0(x^*(t)) + \sum_{i=1}^{m} \frac{1}{-f_i(x^*(t))}\nabla f_i(x^*(t)) + A^\intercal\hat{\nu}. \end{aligned} \tag{5}$$

Lemma 1 (Dual Points from the Central Path)**.** *Every central point yields a dual feasible point, and hence a lower bound on the optimal value, p^*.*

Proof: Define

$$\begin{aligned} \lambda_i^*(t) &= \frac{-1}{tf_i(x^*(t))},\ i = 1, \ldots, m, \\ \nu^*(t) &= \frac{\hat{\nu}}{t}. \end{aligned}$$

We will show that $(\lambda^*(t), \nu^*(t))$ is dual feasible. Since $f_i(x^*(t)) < 0,\ i = 1, \ldots, m$, we have $\lambda^*(t) > 0$. Rewriting (5):

$$\nabla f_0(x^*(t)) + \sum_{i=1}^{m} \lambda_i^*(t)\nabla f(x^*(t)) + A^\intercal\nu^*(t) = 0.$$

Therefore, $x^*(t)$ minimizes the Lagrangian,

$$\begin{aligned} \mathcal{L}(x, \lambda, \nu) =& f_0(x) + \sum_{i=1}^{m} f_i(x) + \nu^\intercal(Ax - b), \\ &\text{for } \lambda = \lambda^*(t) \text{ and } \nu = \nu^*(t) \\ \Rightarrow& (\lambda^*(t), \nu^*(t)) \text{ is a dual feasible pair.} \end{aligned}$$

In addition to showing that it is dual feasible, we show it is dual optimal because it minimizes the Lagrangian. To make a note about how close we are to the actual solution, we see the dual function

$$\begin{aligned} g\left(\lambda^*(t), \nu^*(t)\right) &= f_0(x^*(t)) + \sum_{i=1}^{m} \lambda_i^*(t) f_i(x^*(t)) + \nu^{*\intercal}(t)\left(Ax^*(t) - b\right) \\ &= f_0(x^*(t)) - \frac{m}{t}. \end{aligned}$$

$\therefore\ f_0(x^*(t)) - p^* \leq \frac{m}{t}$, i.e. $x^*(t)$ is no more than $^m/_t$ suboptimal. □

(m = number of inequality constraints)

Interpret the above as a continuous deformation of KKT conditions

$$\begin{aligned} &Ax = b, \\ &f_i(x) < 0,\ i = 1, \ldots, m, \\ &\lambda \geq 0, \\ &\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + A^\intercal\nu = 0, \\ &-\lambda_i f_i(x) = \frac{1}{t},\ i = 1, \ldots, m. \end{aligned}$$

Where the $^1/_t$ term (the **complementary slackness condition**) was zero in our previous considerations.

Note Since $x^*(t)$ is ${}^m/_t$ suboptimal, we can take $t = {}^m/_\varepsilon$, where ε is our specified accuracy goal, and solve:

$$\begin{aligned} &\text{minimize} && \left(\frac{m}{\varepsilon}\right) f_0(x) + \Phi(x) \\ &\text{subject to} && Ax = b. \end{aligned} \tag{6}$$

This is not numerically robust, but it does work well in many cases where ε is non-small. There is a requirement that the initial guess is "good" in the sense that we don't want $t \gg 0$ since the Hessian will not be well-behaved.

3 Barrier Method

Also known as **Logarithmic Barrier Method** or **Central Path Following Method**.

Algorithm 1 Logarithmic Barrier Method

Require: Strictly feasible $\bar{x}$

$t := t^{(0)} > 0$ // Step parameter

$\mu > 1$ // Evolution rate of t

$\varepsilon > 0$ // Tolerance

while do

Centering Steps: Compute $x^*(t)$ by solving

$$\left.\begin{aligned} &\text{minimize} && tf_0 + \Phi \\ &\text{subject to} && A\bar{x} = b \end{aligned}\right\} \text{Newton's Method}$$

// These are the **Inner Iterations**

starting at $\bar{x}$

Update: $x \leftarrow x^*(t)$

Stopping Criterion: **if** $\frac{m}{t} < \varepsilon$ **then** *quit*

Increase t: $t \leftarrow \mu t$

end while

3.1 Choices for Alg. 1

- Rule of thumb is to use $\mu \in (10, 20)$.
- ${}^m/_{t^{(0)}}$ to be same order as $f_0(x^{(0)}) - p^*$.
- If dual feasible (λ, ν) are known with duality gap $\eta = f_0(x^{(0)}) - g(\lambda, \nu)$, then take $f^{(0)} = {}^m/_\eta$.

Note Although we won't go into detail, the inner iteration of Alg. 1 does not have to be Newton's Method, as long as it can solve the equality constrained problem. In the case that we use Newton's Method, we can show that it converges in a number of steps proportional to $\log(\varepsilon)$. Also, the

accuracy of the Newton Method depends on a couple of things, in particular, it depends on the Lipschitz constant for the objective function. If we take a large t, the constant blows up and the accuracy suffers due to an unstable Hessian.

3.2 Convergence

The duality gap after the initial centering step and k additional centering steps is

$$\frac{m}{\mu^k t^{(0)}},$$

and the desired accuracy ε is achieved after

$$\left\lceil \frac{\log\left(\frac{m}{\varepsilon t^{(0)}}\right)}{\log(\mu)} \right\rceil \text{ steps.}$$

A popular piece of software to deal with such convex optimization problems is `CVX` by Grant and Boyd [1, 2]. The freely distributed version is a Matlab implementation that works on general cases. One could either apply for a professional license(free for academia), or streamline the code for their specific application to realize significant speed increases.

Example: SDP

$$\begin{array}{ll} \text{minimize} & c^\intercal x \\ \text{subject to} & x_1F_1 + \ldots + x_nF_n + G \preceq 0 \\ \text{where} & F_1, \ldots, F_n, G \in S^k \end{array}$$

Note Recall that the notation S^k means that the S is considered a $k \times k$ square symmetric matrix.

Lagrangian – Introduce $Z \in S^k$ as a dual variable, then

$$\begin{aligned} \mathcal{L}(x, Z) &= c^\intercal x + \operatorname{trace}\left((x_1F_1 + \ldots + x_nF_n + G)Z\right) \\ &= x_1(c_1 + \operatorname{trace}(F_1Z)) + \ldots + x_n(c_n + \operatorname{trace}(F_nZ)) + \operatorname{trace}(GZ). \end{aligned}$$

This Lagrangian will be affine in x, we can further claim that the **dual function** will be:

$$g(Z) = \inf_x \mathcal{L}(x, Z) = \begin{cases} \operatorname{trace}(GZ),\ \operatorname{trace}(F_iZ) + c_i = 0,\ i \in (1, m) \\ -\infty,\ \text{otherwise} \end{cases}$$

Finally, we have the **dual problem**, which is defined as:

$$\begin{array}{ll} \text{maximize} & \operatorname{trace}(GZ) \\ \text{subject to} & \operatorname{trace}(F_iZ) + c_i = 0,\ i = 1, \ldots, n \\ & Z \succeq 0 \end{array}$$

We have used the fact that S_+^k is self-dual, $k \times k$ symmetric, and positive semi-definite to reach this conclusion. The trace function comes from the idea that: in the space of symmetric matrices, the standard dot-product is essentially a trace.

References

[1] Michael Grant and Stephen Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 3.0 beta. `http://cvxr.com/cvx/`, March 2014.

[2] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. *Recent Advances in Learning and Control (a tribute to M. Vidyasagar)*, V. Blondel, S. Boyd, H. Kimura, editors, pages 95–110, Lecture Notes in Control and Information Sciences, Springer, 2008. `http://stanford.edu/~boyd/graph_dcp.html`