Overview of Dual Ascent

Illinois Institute of Technology
Department of Applied Mathematics

Adam Rumpf
arumpf@hawk.iit.edu

February 19, 2019

Duality Background

Consider a general optimization problem of the form

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & h_i(\mathbf{x}) \leq 0 \qquad i = 1, \ldots, m \\ & \ell_j(\mathbf{x}) = 0 \qquad j = 1, \ldots, r \end{aligned} \tag{P}$$

This is our *primal problem*. For now we will make no assumptions about the objective function f and the constraint functions h_i and ℓ_j. They may or may not be linear, they may or may not be differentiable, and they may or may not be convex.

In general, a *dual problem* of this primal problem is an optimization problem for which any dual objective forms a lower bound for any primal objective. There are many possible choices of dual problem, but most often this term refers to the *Lagrangian dual*, which is based on the Lagrangian function

$$\mathcal{L}(\mathbf{x}; \mathbf{u}, \mathbf{v}) := f(\mathbf{x}) + \sum_{i=1}^{m} u_i h_i(\mathbf{x}) + \sum_{j=1}^{r} v_j \ell_j(\mathbf{x}), \qquad \mathbf{u} \geq \mathbf{0}$$

This function comes from the Lagrangian relaxation of the primal problem, and $\mathbf{u}$ and $\mathbf{v}$ are the unit penalty costs. For any $\mathbf{u} \geq \mathbf{0}$ and for any $\mathbf{v}$, we have $f(\mathbf{x}) \geq \mathcal{L}(\mathbf{x}; \mathbf{u}, \mathbf{v})$, thus the Lagrangian forms a lower bound for the primal objective. If we want to obtain the tightest possible bound, then our goal is to maximize the Lagrangian, which gives us the Lagrangian dual problem

$$\begin{aligned} \max_{\mathbf{u},\mathbf{v}} \quad & g(\mathbf{u}, \mathbf{v}) \\ \text{s.t.} \quad & \mathbf{u} \geq \mathbf{0} \end{aligned} \tag{D}$$

where $g(\mathbf{u}, \mathbf{v}) := \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \mathbf{u}, \mathbf{v})$. It is also possible to write both the primal and the dual in terms of $\mathcal{L}$, by

$$\min_{\mathbf{x}} \max_{\mathbf{u} \geq \mathbf{0}, \mathbf{v}} \quad \mathcal{L}(\mathbf{x}; \mathbf{u}, \mathbf{v}) \tag{P}$$

$$\max_{\mathbf{u} \geq \mathbf{0}, \mathbf{v}} \min_{\mathbf{x}} \quad \mathcal{L}(\mathbf{x}; \mathbf{u}, \mathbf{v}) \tag{D}$$

We will use f^* and g^* to denote the optimal primal and dual objectives, respectively. The *weak duality* property says that $f^* \geq g^*$, which is true even for non-convex problems. For certain classes of problem we also have *strong duality*, in which case $f^* = g^*$ (that is, the duality gap is zero). In particular, strong duality holds for primal problems for which f is convex and *Slater's condition* holds. Slater's condition is that there exist at least one strictly feasible solution, i.e. $\mathbf{x}$ such that all equality constraints hold ($\ell_j(\mathbf{x}) = 0$ for all j) and all inequality constraints hold with strict inequality ($h_i(\mathbf{x}) < 0$ for all i). Note that we only require strict inequality for non-affine h_i.

The *duality gap* is the difference between the primal and dual objectives. For any primal feasible $\mathbf{x}$ and dual feasible $\mathbf{u}, \mathbf{v}$, the duality gap is defined as $f(\mathbf{x}) - g(\mathbf{u}, \mathbf{v})$. The duality gap forms an upper bound for $f(\mathbf{x}) - f^*$, so if we have a duality gap of zero, we know that $\mathbf{x}$ is the optimal primal solution. This property forms the foundation of many solution techniques. In some cases we can attempt to find algebraic solutions to $f(\mathbf{x}) - g(\mathbf{u}, \mathbf{v}) = 0$. In other cases we may use an iterative technique and use $f(\mathbf{x}) - g(\mathbf{u}, \mathbf{v}) \leq \epsilon$ as a stopping criterion.

Review of Subgradient

The *subgradient* is a generalization of the classical *gradient*. While the gradient only exists for differentiable functions, the subgradient exists even at non-differentiable points. Intuitively, if we imagine f as a surface in 3D space, then the gradient at any given point is given by finding a plane tangent to that point and finding the direction of steepest ascent on that plane. If f is non-differentiable, then it may have some "kinks" that prevent us from defining a single unambiguous tangent plane. However, even at these kinks there will always be a set of planes that pass through the specified point and lie completely underneath the surface (locally). The set of gradients of all such planes is called the subgradient at that point.

There are some relatively straightforward limit-based definitions for the subgradient which generalize the limit definition of the gradient, but within mathematical optimization there is an equivalent inequality-based definition which is generally more useful. A subgradient of $f : U \to \mathbb{R}$ at a point $\mathbf{x}_0$ is any vector $\mathbf{v}$ satisfying

$$f(\mathbf{x}) - f(\mathbf{x}_0) \geq \mathbf{v} \cdot (\mathbf{x} - \mathbf{x}_0) \qquad \forall \mathbf{x} \in U$$

The set of all subgradients at $\mathbf{x}_0$ is called the *subdifferential* at $\mathbf{x}_0$, and is denoted $\partial f(\mathbf{x}_0)$. Note that, if f is differentiable at $\mathbf{x}_0$, then the subdifferential will consist entirely of a single value, and that value is simply the classical derivative of f at $\mathbf{x}_0$.

The easiest example to visualize is the absolute value function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = |x|$. At any point $x_0 < 0$, f is differentiable (with derivative -1), and at any point $x_0 > 0$ it is differentiable (with derivative 1), but f is not differentiable at $x_0 = 0$. However there are a lot of lines that we could draw through the point $(0, 0)$ that would lie completely underneath the curve. In particular we could draw any line passing through the origin with a slope between -1 and 1, and so the subdifferential of f at $x_0 = 0$ is the interval $[-1, 1]$.

There are many concepts from calculus and optimization that can be generalized to non-differentiable functions by using the subdifferential instead of the derivative. For example, for a differentiable function f a necessary condition for local optimality at $\mathbf{x}_0$ is that the gradient be zero, i.e. $\nabla f(\mathbf{x}_0) = 0$. For non-differentiable f, the necessary condition is that the subgradient contain zero, i.e. $0 \in \partial f(\mathbf{x}_0)$. Intuitively this implies that it is possible to draw a level plane at $(\mathbf{x}_0, f(\mathbf{x}_0))$ that does not intersect the surface, which could occur at a peak or a valley or a ridge.

Dual Ascent

Dual ascent takes advantage of the fact that the dual problem is always convex, and so we can apply techniques from convex minimization. Specifically, we use gradient (or subgradient) ascent on the dual variables. The idea is to start at an initial guess, take a small step in the direction of the gradient, and repeat. As long as the step sizes are chosen correctly, this will converge to a local maximum. The subgradient version is similar, except that the direction may now be chosen to be any vector in the subgradient.

The dual subgradient ascent method is as follows:

1. Choose an initial dual guess $\mathbf{u}^{(0)} \geq \mathbf{0}, \mathbf{v}^{(0)}$.
2. Repeat the following, where $k = 1, 2, \ldots$ represents the step number.
   (a) Update the step size t_k.
   (b) Update the primal solution $\mathbf{x}^{(k)} \in \operatorname{argmin}_{\mathbf{x}} \mathcal{L}(\mathbf{x}; \mathbf{u}^{(k-1)}, \mathbf{v}^{(k-1)})$.
   (c) Update the dual variables $\mathbf{u}^{(k)} = \max\{\mathbf{u}^{(k-1)} + t_k h(\mathbf{x}^{(k)}), \mathbf{0}\}$, $\mathbf{v}^{(k)} = \mathbf{v}^{(k-1)} + t_k \ell(\mathbf{x}^{(k)})$.

To explain, the primal solution update is based on the fact that, for the "correct" choices of penalty cost $\mathbf{u}$ and $\mathbf{v}$, the primal optimum $\mathbf{x}$ will minimize the Lagrangian relaxation. This step involves minimizing an unconstrained

function, which can generally be done fairly quickly. The dual updates are simply the standard hill climb updates, and come from the fact that $\nabla_{\mathbf{u}}\mathcal{L} = \mathbf{h}$ and $\nabla_{\mathbf{v}}\mathcal{L} = \boldsymbol{\ell}$. For reasons of dual feasibility we must also always ensure that $\mathbf{u} \geq \mathbf{0}$.

This always works if g is differentiable, but not necessarily if it is not, since a subgradient is not necessarily always a direction of ascent. To compensate for this, whenever we update $\mathbf{u}$ and $\mathbf{v}$, we always retain the option of replacing the updated versions with a previous version by keeping the best known value of g in memory.

As for the step sizes t_k, it is possible to use a constant step size, which can work for generating an approximate solution. However, convergence is only guaranteed if the step sizes approach zero, but not "too fast". Specifically, they must satisfy $\sum_{k=1}^{\infty} t_k^2 < \infty$ and $\sum_{k=1}^{\infty} t_k = \infty$.

Dual ascent is most effective when the Lagrangian relaxation (which is unconstrained) can be easily minimized for any fixed set of penalty costs. In particular this is the case for linear programs, although specialized algorithms for particular problem structures are generally more efficient. If the primal problem can be decomposed into a block structure, then the dual ascent algorithm can be parallelized as explained in the next section.

Dual Decomposition

Consider a primal problem of a particular block structure. Specifically, suppose there are B blocks of variables $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_B)$, and for each block we have a separate cost $f_i(\mathbf{x}_i)$ and constraint $h_{ij}(\mathbf{x}_i) \leq 0$, so that

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^{B} f_i(\mathbf{x}_i) \\ \text{s.t.} \quad & \sum_{i=1}^{B} h_{ij}(\mathbf{x}_i) \leq 0 \qquad j = 1, \ldots, m \end{aligned} \tag{P}$$

Since the Lagrangian $\mathcal{L}$ in this case can be decomposed into a sum over the B blocks, so too can its subgradient, and so rather than trying to minimize all of $\mathcal{L}$ at once, we can instead minimize each component of $\mathcal{L}$ separately. That is, instead of trying to find

$$\mathbf{x} \in \operatorname*{argmin}_{\mathbf{x}} \sum_{i=1}^{B} \left(f_i(\mathbf{x}) + \sum_{j=1}^{m} u_j h_{ij}(\mathbf{x}_i) \right)$$

we can separately find

$$\mathbf{x}_i \in \operatorname*{argmin}_{\mathbf{x}_i} f_i(\mathbf{x}_i) + \sum_{j=1}^{m} u_j h_{ij}(\mathbf{x}_i)$$

for $i = 1, \ldots, B$, which can be done in parallel since all of these problems are independent. In each iteration of dual ascent, we use separate processors to solve each of the B minimization problems to find $\mathbf{x}_i^{(k)}$ based on $\mathbf{u}^{(k-1)}$, and then we use all collected values of $\mathbf{x}_i^{(k)}$ to update $\mathbf{u}^{(k)}$.

References

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.