University of California, Berkeley Lecturer: Benjamin Recht

**CS227C/STAT260 Convex Optimization and Approximation: Optimization for Modern Data Analysis**

March 8, 2016 Notes: Benjamin Recht

---

# Lecture 13: Dual Decomposition

## 1 Dual Ascent

Let's again begin with an equality constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega \\ & Ax = b \end{array} \tag{1}$$

Here, assume that $\Omega$ is a closed convex set, $f$ is convex and differentiable, and $A$ is full rank.

The Lagrangian for this problem is given by the function

$$\mathcal{L}(x, \lambda) = f(x) + \lambda^T (Ax - b)\,.$$

As we showed last time, problem (1) is equivalent to the optimization problem

$$\min_{x \in \Omega} \max_{\lambda \in \mathbb{R}^n} f(x) + \lambda^T (Ax - b)\,.$$

The *dual problem* associated with (1) is

$$\max_{\lambda \in \mathbb{R}^n} \min_{x \in \Omega} f(x) + \lambda^T (Ax - b)$$

And the concave function

$$q(\lambda) := \min_{x \in \Omega} f(x) + \lambda^T (Ax - b)$$

is called the *dual function* associated with problem (1).

Recall again from last lecture that if the relative interior of $\Omega$ contains a point satisfying the equality constraint then *strong duality* holds. In particular, if we can solve the dual problem, then we can find a primal optimal point by solving the problem

$$\text{minimize}_{x \in \Omega} \max_{\lambda \in \mathbb{R}^n} f(x) + \lambda_\star^T (Ax - b) \tag{2}$$

where $\lambda_\star$ is a dual optimal solution.

Note that even if $\lambda$ is only approximately dual optimal, solving (2) gives a reasonable approximation to the original optimization problem. This can be seen by the calculation

$$\begin{aligned} f(x_\star) = q(\lambda_\star) &\leq q(\lambda) + \epsilon \\ &= \inf_{x \in \Omega} \mathcal{L}(x, \lambda) + \epsilon \\ &\leq \mathcal{L}(x, \lambda) + \epsilon \\ &= f(x) + \lambda^T (Ax - b) + \epsilon \end{aligned}$$

Hence, if $\|Ax - b\|$ is small and our dual optimal solution is nearly accurate, then we get a reasonable approximation to the optimal function value.

## 2 Algorithms

So how do we solve the dual problem? It's concave, so we can apply the subgradient method:

$$x^{(k)} \leftarrow \arg\min_{x\in\Omega} \mathcal{L}(x, \lambda^{(k)}$$
$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} + s_k(Ax^{(k)} - b)$$

Using our analysis, we then have

$$\frac{1}{\sum_{k=1}^{T} s_k} \sum_{k=1}^{T} s_k \lambda_k \to \lambda_\star$$

at a rate of $O(T^{-1/2})$.

To get a $1/T$ rate, we can do something a bit more clever. Note that we can apply the proximal-point method to the dual problem. This would result in the iteration

$$\lambda^{(k+1)} \leftarrow \arg\max_{\lambda} \inf_{x\in\Omega} f(x) + \lambda^T(Ax - b) - \frac{1}{2\alpha_k}\|\lambda - \lambda^{(k)}\|^2$$

Now, the objective is convex in $x$ and strongly convex in $\lambda$. So we can swap the infimum and supremum by Sion's minimax theorem. This results in the equivalent problem

$$\inf_{x\in\Omega} \left\{ \max_{\lambda} f(x) + \lambda^T(Ax - b) - \frac{1}{2\alpha_k}\|\lambda - \lambda^{(k)}\|^2 \right\}$$

But the internal problem here has a trivial solution with respect to $\lambda$:

$$\lambda^{(k+1)} = \lambda^{(k)} + \alpha_k(Ax - b)$$

Plugging this solution back in for $\lambda$ shows that the optimal $x$ can be found by

$$\inf_{x\in\Omega} f(x) + \lambda^{(k)}(Ax - b) + \frac{\alpha_k}{2}\|Ax - b\|^2 =: \mathcal{L}_{\alpha_k}(x, \lambda^{(k)})\,.$$

That is, we find the next $x$ by minimizing an *Augmented Lagrangian*.

This results in the iteration

$$x^{(k)} \leftarrow \arg\min_{x\in\Omega} \mathcal{L}_{\alpha_k}(x, \lambda^{(k)})$$
$$\lambda^{(k+1)} \leftarrow \lambda^{(k)} + \alpha_k(Ax^{(k)} - b)$$

Note that this algorithm is *nearly identical* to the subgradient method. The only difference is that we have to solve an augmented Lagrangian for the $x$-step. This can speed up iterations, but also may add algorithmic difficulty when computing the $x$-step.

### 2.1 Practical method of multipliers

Note that the proximal-point method is guaranteed to converge for even a constant step-size $\alpha_k$. Nonetheless, there are some heuristics that seem to work very well in practice. In particular, the Lancelot code for nonlinear programming makes the following recommendations:

The code has two parameters $\eta \in (0, 1)$ and $\gamma > 1$. We perform the following steps

1. Start with a small $\alpha_0$.
2. After minimizing your augmented Lagrangian at step $k$, compute $\delta = \|Ax^{(k)} - b\|^2$.
3. If $\delta < \eta\delta_k$, our iterate became more feasible. In this case, we don't need to increase the penalty in the augmented Lagrangian. Thus, we proceed as
   (a) $\lambda^{(k+1)} \leftarrow \lambda^{(k)} + \alpha_k(Ax^{(k)} - b)$
   (b) $\alpha_{k+1} \leftarrow \alpha_k$
   (c) $\delta_{k+1} \leftarrow \delta$
4. If $\delta \geq \eta\delta_k$, then we didn't improve the feasibility of $x$. So we increase the penalty parameter and try again:
   (a) $\lambda^{(k+1)} \leftarrow \lambda^{(k)}$
   (b) $\alpha_{k+1} \leftarrow \gamma\alpha_k$
   (c) $\delta_{k+1} \leftarrow \delta_k$

Typical values for $\eta$ are $1/4$ and for $\gamma$ is 10.

# 3 Examples

## 3.1 Consensus Optimization

Let $G = (V, E)$ be a graph and consider the optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{v\in V} f_v(x_v) \\ \text{subject to} & x_u = x_v \qquad (u,v) \in E \end{array} \tag{3}$$

The agumented Lagrangian is

$$\begin{aligned} \mathcal{L}(x,\lambda) &= \sum_{v\in V} f_v(x_v) + \sum_{(u,v)\in E} \lambda_{u,v}^T(x_u - x_v) \\ &= \sum_{v\in V} \left\{ f_v(x_v) + \left(\sum_{(v,w)\in E} \lambda_{v,w} - \sum_{(u,v)\in E} \lambda_{u,v}\right)^T x_v \right\}. \end{aligned}$$

Note that the $x$-step is now completely distributed. We can compute the minimizer with respect to $x_v$ solely by knowing the Lagrange multipliers associated with the neighbors of $v$ in $G$. The $\lambda$-step consists of the simple step

$$\lambda_{u,v} = \lambda_{u,v} + s(x_u - x_v)$$

Note that there are many decompositions of this form. Suppose we want to minimize $\sum_{v\in V} f_v(x)$. Here, some of the $f_v$ might even be indicator functions of convex sets. We can rewrite this optimization problem to decouple the constraints in the from (3). This is completely equivalent to the original formulation, no matter what graph we choose, provided the graph is connected.

Note that if we used the augmented Lagrangian, this distributed decoupling would not work, as we cannot split the quadratic term. We'll discuss how to address this in the next lecture.

### 3.2 Utility Maximization

The general utility maximization problem is stated as

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} U_i(x_i) \\ \text{subject to} & Rx \leq c \end{array}$$

Here, each utility function represents some happiness or well-being for the $i$th user as a function of the amount of resource $x_i$. The inequalities are resource constraints, coupling the amount of utility available to each user.

Lets rewrite this in our standard form

$$\begin{array}{ll} \text{minimize} & -\sum_{i=1}^{n} U_i(x_i) \\ \text{subject to} & Rx - c + s = 0 \\ & s \geq 0 \end{array}$$

Then our Lagrangian becomes

$$\mathcal{L}(x, s, \lambda) = \sum_{i=1}^{n} -U_i(x_i) + \lambda^T (Rx - c + s)$$

Minimizing with respect to $s \geq 0$ shows that $\lambda \geq 0$. Otherwise, the Lagrangian is unbounded below. Thus, our dual problem is equivalent to

$$\max_{\lambda \geq 0} \min_{x} \sum_{i=1}^{n} -U_i(x_i) + \lambda^T (Rx - c)$$

The $x$-step can be done in a distributed fashion with each user maximizing

$$U(x_i) - \left[ \sum_{j=1}^{p} R_{ji} \lambda_j \right] x_i$$

Then we can update $\lambda$ by the rule

$$\lambda \leftarrow [\lambda + \alpha(Rx - c)]_+ \ .$$

The dynamics of this model are very interesting. $\lambda$ can be interpreted as *prices* for a particular resource. If the prices are large, users get negative cost for acquiring more of their quantity $x_i$. If the resource constraints are loose, then the prices go down. If they are violated, then the prices go up.

### 3.3 Linear and Quadratic Programming

Consider the problem

$$\begin{array}{ll} \text{minimize} & c^T x + \frac{1}{2} x^T Q x \\ \text{subject to} & \ell \leq x \leq u \\ & Ax = b \end{array}$$

That is, we aim to solve a box-constrained quadratic program. Note that if $\ell = 0$ and $u = \inf$ and $Q = 0$, this is a standard form linear program.

The augmented Lagrangian for this problem is

$$\mathcal{L}(x, \lambda) = c^T x + \tfrac{1}{2} x^T Q x + \lambda^T (Ax - b) + \tfrac{\alpha_k}{2} \|Ax - b\|^2$$

and the $x$-step reduces to

$$\text{minimize}_{\ell \leq x \leq u} c^T x + \tfrac{1}{2} x^T Q x + \lambda^T (Ax - b) + \tfrac{\alpha_k}{2} \|Ax - b\|^2$$

which you could solve via the projected gradient method, or something similar.