Dual Methods

Lecturer: Ryan Tibshirani
Convex Optimization 10-725/36-725

Last time: proximal Newton method

Consider the problem

$$\min_x \; g(x) + h(x)$$

where g, h are convex, g is twice differentiable, and h is "simple". Proximal Newton method: let $x^{(0)} \in \mathbb{R}^n$, and repeat:

$$v^{(k)} = \operatorname*{argmin}_v \; \nabla g(x^{(k-1)})^T v + \frac{1}{2} v^T \nabla^2 g(x^{(k-1)}) v + h(x^{(k-1)} + v)$$

$$x^{(k)} = x^{(k-1)} + t_k v^{(k)}, \quad k = 1, 2, 3, \ldots$$

Step sizes are typically chosen by backtracking

- Iterations here are typically very expensive (computing $d^{(k)}$ is typically a formidable task)
- But typically very few iterations are needed until convergence: under appropriate conditions, get local quadratic convergence

Reminder: conjugate functions

Recall that given $f : \mathbb{R}^n \to \mathbb{R}$, the function

$$f^*(y) = \max_x \; y^T x - f(x)$$

is called its conjugate

- Conjugates appear frequently in dual programs, since

$$-f^*(y) = \min_x \; f(x) - y^T x$$

- If f is closed and convex, then $f^{**} = f$. Also,

$$x \in \partial f^*(y) \iff y \in \partial f(x) \iff x \in \operatorname*{argmin}_z \; f(z) - y^T z$$

- If f is strictly convex, then $\nabla f^*(y) = \operatorname*{argmin}_z \; f(z) - y^T z$

Proof details

We will show that $x \in \partial f^*(y) \iff y \in \partial f(x)$, assuming that f is convex and closed

Proof of "$\Longleftarrow$": Suppose $y \in \partial f(x)$. Then $x \in M_y$, the set of maximizers of $y^T z - f(z)$ over z. But $f^*(y) = \max_z \; y^T z - f(z)$ and $\partial f^*(y) = \mathrm{cl}(\mathrm{conv}(\cup_{z \in M_y} \{z\}))$. Thus $x \in \partial f^*(y)$.

Proof of "$\Longrightarrow$": From what we showed above, if $x \in \partial f^*(y)$, then $y \in \partial f^{**}(x)$, but $f^{**} = f$.

Clearly $y \in \partial f(x) \iff x \in \mathrm{argmin}_z \; f(z) - y^T z$

Lastly if f is strictly convex, then we know that $f(z) - y^T z$ has a unique minimizer over z, and this must be $\nabla f^*(y)$

Outline

Today:

- Dual (sub)gradient methods
- Dual decomposition
- Augmented Lagrangians
- A peak at ADMM

Dual (sub)gradient methods

Even if we can't derive dual (conjugate) in closed form, we can still use dual-based subgradient or gradient methods

Example: consider the problem

$$\min_x \ f(x) \ \text{ subject to } \ Ax = b$$

Its dual problem is

$$\max_u \ -f^*(-A^T u) - b^T u$$

where f^* is conjugate of f. Defining $g(u) = -f^*(-A^T u) - b^T u$, note that

$$\partial g(u) = A\partial f^*(-A^T u) - b$$

Therefore, using what we know about conjugates

$$\partial g(u) = Ax - b \quad \text{where} \quad x \in \operatorname*{argmin}_z \; f(z) + u^T Az$$

The dual subgradient method (for maximizing the dual objective) starts with an initial dual guess $u^{(0)}$, and repeats for $k = 1, 2, 3, \ldots$

$$x^{(k)} \in \operatorname*{argmin}_x \; f(x) + (u^{(k-1)})^T Ax$$
$$u^{(k)} = u^{(k-1)} + t_k(Ax^{(k)} - b)$$

Step sizes t_k, $k = 1, 2, 3, \ldots$, are chosen in standard ways

Recall that if f is strictly convex, then f^* is differentiable, and so this becomes dual gradient ascent, which repeats for $k = 1, 2, 3, \ldots$

$$x^{(k)} = \operatorname*{argmin}_x \; f(x) + (u^{(k-1)})^T Ax$$
$$u^{(k)} = u^{(k-1)} + t_k(Ax^{(k)} - b)$$

(Difference is that each $x^{(k)}$ is unique, here.) Again, step sizes t_k, $k = 1, 2, 3, \ldots$ are chosen in standard ways

Also, proximal gradients and acceleration can be applied as they would usually

Lipschitz gradients and strong convexity

Assume that f is a closed and convex function. Then f is strongly convex with parameter $d \iff \nabla f^*$ Lipschitz with parameter $1/d$

Proof of "$\Longrightarrow$": Recall, if g strongly convex with minimizer x, then

$$g(y) \geq g(x) + \frac{d}{2}\|y - x\|_2, \quad \text{for all } y$$

Hence defining $x_u = \nabla f^*(u)$, $x_v = \nabla f^*(v)$,

$$f(x_v) - u^T x_v \geq f(x_u) - u^T x_u + \frac{d}{2}\|x_u - x_v\|_2^2$$

$$f(x_u) - v^T x_u \geq f(x_v) - v^T x_v + \frac{d}{2}\|x_u - x_v\|_2^2$$

Adding these together, using Cauchy-Schwartz, and rearranging shows that $\|x_u - x_v\|_2 \leq \cdot\|u - v\|_2/d$

Convergence guarantees

The following results hold from combining the last fact with what we already know about gradient descent:

- If f is strongly convex with parameter d, then dual gradient ascent with fixed step sizes $t_k = d$, $k = 1, 2, 3, \ldots$, converges at the rate $O(1/\epsilon)$
- If f is strongly convex with parameter d, and ∇f is Lipschitz with parameter L, then dual gradient ascent with step sizes $t_k = 2/(1/d + 1/L)$, $k = 1, 2, 3, \ldots$, converges at the rate $O(\log(1/\epsilon))$

Note that these results describe convergence of the dual objective to its optimal value

Dual decomposition

Consider

$$\min_x \sum_{i=1}^B f_i(x_i) \quad \text{subject to} \quad Ax = b$$

Here $x = (x_1, \ldots x_B) \in \mathbb{R}^n$ divides into B blocks of variables, with each $x_i \in \mathbb{R}^{n_i}$. We can also partition A accordingly

$$A = [A_1 \ldots A_B], \quad \text{where } A_i \in \mathbb{R}^{m \times n_i}$$

Simple but powerful observation, in calculation of (sub)gradient, is that the minimization decomposes into B separate problems:

$$x^+ \in \operatorname*{argmin}_x \sum_{i=1}^B f_i(x_i) + u^T Ax$$

$$\iff x_i^+ \in \operatorname*{argmin}_{x_i} f_i(x_i) + u^T A_i x_i, \quad i = 1, \ldots B$$

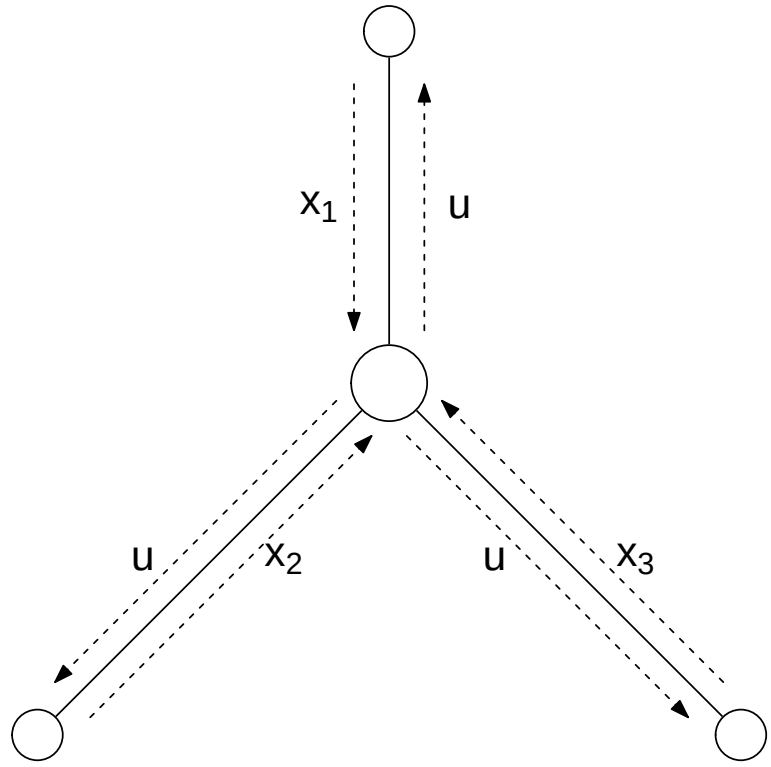
Dual decomposition algorithm: repeat for $k = 1, 2, 3, \ldots$

$$x_i^{(k)} \in \operatorname*{argmin}_{x_i} \; f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots B$$

$$u^{(k)} = u^{(k-1)} + t_k \left(\sum_{i=1}^{B} A_i x_i^{(k)} - b \right)$$

Can think of these steps as:

- Broadcast: send u to each of the B processors, each optimizes in parallel to find x_i
- Gather: collect $A_i x_i$ from each processor, update the global dual variable u



Dual decomposition with inequality constraints

Consider

$$\min_x \sum_{i=1}^B f_i(x_i) \quad \text{subject to} \quad \sum_{i=1}^B A_i x_i \leq b$$

Dual decomposition (projected subgradient method): repeat for $k = 1, 2, 3, \ldots$

$$x_i^{(k)} \in \operatorname*{argmin}_{x_i} \; f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots B$$

$$u^{(k)} = \left(u^{(k-1)} + t_k \left(\sum_{i=1}^B A_i x_i^{(k)} - b \right) \right)_+$$

where u_+ denotes the positive part of u, i.e., $(u_+)_i = \max\{0, u_i\}$, $i = 1, \ldots, m$

Price coordination interpretation (Vandenberghe):

- Have B units in a system, each unit chooses its own decision variable x_i (how to allocate its goods)
- Constraints are limits on shared resources (rows of A), each component of dual variable u_j is price of resource j
- Dual update:

$$u_j^+ = (u_j - ts_j)_+, \quad j = 1, \ldots m$$

  where $s = b - \sum_{i=1}^{B} A_i x_i$ are slacks

  - Increase price u_j if resource j is over-utilized, $s_j < 0$
  - Decrease price u_j if resource j is under-utilized, $s_j > 0$
  - Never let prices get negative

Augmented Lagrangian method

also known as: method of multipliers

Disadvantage of dual ascent: require strong conditions to ensure convergence. Improved by augmented Lagrangian method, also called method of multipliers. We transform the primal problem:

$$\begin{array}{ll} \min\limits_{x} & f(x) + \dfrac{\rho}{2}\|Ax - b\|_2^2 \\ \text{subject to} & Ax = b \end{array}$$

where $\rho > 0$ is a parameter. Clearly equivalent to original problem, and objective is strongly convex when A has full column rank. Use dual gradient ascent: repeat for $k = 1, 2, 3, \ldots$

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_x \; f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2}\|Ax - b\|_2^2 \\ u^{(k)} &= u^{(k-1)} + \rho(Ax^{(k)} - b) \end{aligned}$$

Notice step size choice $t_k = \rho$, $k = 1, 2, 3, \ldots$ in dual algorithm. Why? Since $x^{(k)}$ minimizes $f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2}\|Ax - b\|_2^2$ over x, we have

$$\begin{aligned} 0 &\in \partial f(x^{(k)}) + A^T\left(u^{(k-1)} + \rho(Ax^{(k)} - b)\right) \\ &= \partial f(x^{(k)}) + A^T u^{(k)} \end{aligned}$$

This is the stationarity condition for the original primal problem; can show under mild conditions that $Ax^{(k)} - b$ approaches zero (i.e., primal iterates approach feasibility), hence in the limit KKT conditions are satisfied and $x^{(k)}, u^{(k)}$ approach optimality

Advantage: much better convergence properties. Disadvantage: lose decomposability! (Separability is compromised by augmented Lagrangian ...)

Alternating direction method of multipliers

Alternating direction method of multipliers or ADMM: the best of both worlds, i.e., we get strong convergence properties, along with decomposability. Consider

$$\min_{x,z} \; f(x) + g(z) \;\; \text{subject to} \;\; Ax + Bz = c$$

As before, we augment the objective

$$\begin{aligned} \min_{x} \quad & f(x) + g(z) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2 \\ \text{subject to} \quad & Ax + Bz = c \end{aligned}$$

for a parameter $\rho > 0$. We define augmented Lagrangian

$$L_\rho(x, z, u) = f(x) + g(z) + u^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2$$

ADMM repeats the steps, for $k = 1, 2, 3, \ldots$

$$
\begin{aligned}
x^{(k)} &= \operatorname*{argmin}_x \; L_\rho(x, z^{(k-1)}, u^{(k-1)}) \\
z^{(k)} &= \operatorname*{argmin}_z \; L_\rho(x^{(k)}, z, u^{(k-1)}) \\
u^{(k)} &= u^{(k-1)} + \rho(Ax^{(k)} + Bz^{(k)} - c)
\end{aligned}
$$

Note that the usual method of multipliers would have replaced the first two steps by a joint minimization

$$
(x^{(k)}, z^{(k)}) = \operatorname*{argmin}_{x,z} \; L_\rho(x, z, u^{(k-1)})
$$

Convergence guarantees

Under modest assumptions on f, g (these do not require A, B to be full rank), the ADMM iterates satisfy, for any $\rho > 0$:

- Residual convergence: $r^{(k)} = Ax^{(k)} - Bz^{(k)} - c \to 0$ as $k \to \infty$, i.e., primal iterates approach feasibility
- Objective convergence: $f(x^{(k)}) + g(z^{(k)}) \to f^\star + g^\star$, where $f^\star + g^\star$ is the optimal primal objective value
- Dual convergence: $u^{(k)} \to u^\star$, where $u^\star$ is a dual solution

For details, see Boyd et al. (2010). Note that we do not generically get primal convergence, but this is true under more assumptions

Convergence rate: not known in general, theory is currently being developed, e.g., in Hong and Luo (2012), Deng and Yin (2012), Iutzeler et al. (2014), Nishihara et al. (2015). Roughly, it behaves like a first-order method (or a bit faster)

ADMM in scaled form

It is often easier to express the ADMM algorithm in a scaled form, where we replace the dual variable u by a scaled variable $w = u/\rho$. In this parametrization, the ADMM steps are:

$$
\begin{aligned}
x^{(k)} &= \underset{x}{\operatorname{argmin}} \; f(x) + \frac{\rho}{2}\|Ax + Bz^{(k-1)} - c + w^{(k-1)}\|_2^2 \\
z^{(k)} &= \underset{z}{\operatorname{argmin}} \; g(z) + \frac{\rho}{2}\|Ax^{(k)} + Bz - c + w^{(k-1)}\|_2^2 \\
w^{(k)} &= w^{(k-1)} + Ax^{(k)} + Bz^{(k)} - c
\end{aligned}
$$

Note that here the kth iterate $w^{(k)}$ is just given by a running sum of residuals:

$$
w^{(k)} = w^{(0)} + \sum_{i=1}^{k} \left(Ax^{(i)} + Bz^{(i)} - c\right)
$$

Example: alternating projections

Consider finding a point in intersection of convex sets $C, D \subseteq \mathbb{R}^n$:

$$\min_x \; I_C(x) + I_D(x)$$

To get this into ADMM form, we express it as

$$\min_{x,z} \; I_C(x) + I_D(z) \;\; \text{subject to} \;\; x - z = 0$$

Each ADMM cycle involves two projections:

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_x \; P_C\left(z^{(k-1)} - w^{(k-1)}\right) \\ z^{(k)} &= \operatorname*{argmin}_z \; P_D\left(x^{(k)} + w^{(k-1)}\right) \\ w^{(k)} &= w^{(k-1)} + x^{(k)} - z^{(k)} \end{aligned}$$

Like the classical alternating projections method, but more efficient

References

- S. Boyd and N. Parikh and E. Chu and B. Peleato and J. Eckstein (2010), "Distributed optimization and statistical learning via the alternating direction method of multipliers"
- W. Deng and W. Yin (2012), "On the global and linear convergence of the generalized alternating direction method of multipliers"
- M. Hong and Z. Luo (2012), "On the linear convergence of the alternating direction method of multipliers"
- F. Iutzeler and P. Bianchi and Ph. Ciblat and W. Hachem, (2014), "Linear convergence rate for distributed optimization with the alternating direction method of multipliers"
- R. Nishihara and L. Lessard and B. Recht and A. Packard and M. Jordan (2015), "A general analysis of the convergence of ADMM"
- L. Vandenberghe, Lecture Notes for EE 236C, UCLA, Spring 2011-2012