Dual Ascent

Lecturer: Aarti Singh
Co-instructor: Pradeep Ravikumar

Convex Optimization 10-725/36-725

Algorithms based on dual problem

Since dual problem is always convex (concave maximization) irrespective of primal, we can use the methods for convex minimization we have learnt so far.

Key challenge: Differentiability of Lagrange dual function $g(u, v)$

- Whenever $L(x, u, v)$ is minimized over a unique $x_{u,v}$ for any given (u, v), then g is differentiable.
- This holds, for example, if f is strictly convex and h is affine.
- But in general, this often does not hold. In particular, whenever there is duality gap, the dual function is not differentiable at every dual optimal solution.

Algorithms based on dual problem

Since dual problem is always convex (concave maximization) irrespective of primal, we can use the methods for convex minimization we have learnt so far.

Key challenge: Differentiability of Lagrange dual function $g(u, v)$

- Whenever $L(x, u, v)$ is minimized over a unique $x_{u,v}$ for any given (u, v), then g is differentiable.
- This holds, for example, if f is strictly convex and h is affine.
- But in general, this often does not hold. In particular, whenever there is duality gap, the dual function is not differentiable at every dual optimal solution.

Algorithms for dual problems:

- Differentiable - Dual gradient ascent (next)
- Non-differentiable - Dual subgradient ascent (next), Cutting plane, Decomposition methods

Dual ascent

Since dual problem is always convex (concave maximization) irrespective of primal, we can use gradient or sub-gradient ascent on the dual variables.

Let x' be a minimizer of $L(x, u', v')$ for given $u' \geq 0, v'$. Then

$$\begin{bmatrix} h(x') \\ \ell(x') \end{bmatrix} \text{ is a (sub)gradient of } g \text{ at } \begin{bmatrix} u' \\ v' \end{bmatrix} \text{ because } \forall u, v$$

Dual ascent

Since dual problem is always convex (concave maximization) irrespective of primal, we can use gradient or sub-gradient ascent on the dual variables.

Let x' be a minimizer of $L(x, u', v')$ for given $u' \geq 0, v'$. Then

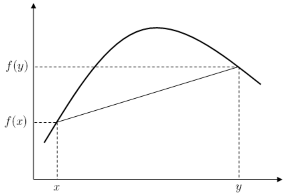
$\begin{bmatrix} h(x') \\ \ell(x') \end{bmatrix}$ is a (sub)gradient of g at $\begin{bmatrix} u' \\ v' \end{bmatrix}$ because $\forall u, v$

$$\begin{aligned}
g(u,v) &= \min_x L(x,u,v) \\
&= \min_x f(x) + u^\top h(x) + v^\top \ell(x) \\
&\leq f(x') + u^\top h(x') + v^\top \ell(x') \\
&= f(x') + u'^\top h(x') + (u-u')^\top h(x') \\
&\quad + v'^\top \ell(x') + (v-v')^\top \ell(x') \\
&= g(u',v') + (u-u')^\top h(x') + (v-v')^\top \ell(x')
\end{aligned}$$

Last step follows since x' is a minimizer of $L(x, u', v')$.

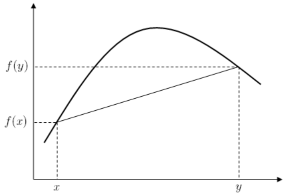
Recall: Subgradient of a *concave* function f at x is any s s.t.

$$f(y) \leq f(x) + s^\top (y - x) \quad \forall y$$



Recall: Subgradient of a *concave* function f at x is any s s.t.

$$f(y) \le f(x) + s^\top (y - x) \quad \forall y$$



Dual (sub)gradient ascent method

- Start
  with an initial dual guess $u^{(0)} \ge 0, v^{(0)}$.
- Repeat for $k = 1, 2, 3, \ldots$

$$x^{(k)} \in \operatorname*{argmin}_x \; f(x) + (u^{(k-1)})^\top h(x) + (v^{(k-1)})^\top \ell(x)$$
$$u^{(k)} = \max\{u^{(k-1)} + t_k h(x^{(k)}), 0\}$$
$$v^{(k)} = v^{(k-1)} + t_k \ell(x^{(k)})$$

Step sizes $t_k, k = 1, 2, 3, \ldots$ are chosen in standard ways
Proximal gradients and acceleration can be applied as they would usually

Method of multipliers as dual ascent

Recall Method of Multipliers: Solve sequence of unconstrained minimization of Augmented Lagrangian

$$x^{(k)} = \arg\min_x \ L_{c^{(k)}}(x, \lambda^{(k)})$$

where for equality constrained problem ($\min_x f(x)$ s.t. $h(x) = 0$)

$$L_{c^{(k)}}(x, \lambda^{(k)}) = f(x) + {\lambda^{(k)}}^\top h(x) + \frac{c^{(k)}}{2} \|h(x)\|^2$$

and using the following multiplier update:

$$\lambda^{(k+1)} = \lambda^{(k)} + c^{(k)} h(x^{(k)}).$$

This is precisely dual ascent for the augmented problem!

Gradient vs Subgradient descent/ascent

- Subgradient may not be a direction of ascent at (u, v) where dual function g is non-differentiable, so we take best iterate so far:

$$g((u^{(k)}, v^{(k)})_{\mathsf{best}}) = \max_{i=0,\dots k} \; g(u^{(i)}, v^{(i)})$$

Gradient vs Subgradient descent/ascent

- Subgradient may not be a direction of ascent at (u, v) where dual function g is non-differentiable, so we take best iterate so far:

$$g((u^{(k)}, v^{(k)})_{\mathsf{best}}) = \max_{i=0,\dots k} \; g(u^{(i)}, v^{(i)})$$

- The subgradient makes an angle < 90 with all ascent directions at (u, v)

$$f(y) \leq f(x) + s^\top (y - x) \quad \forall y \quad \Rightarrow \quad 0 < s^\top (y - x) \quad \forall f(y) > f(x)$$

Gradient vs Subgradient descent/ascent

- Subgradient may not be a direction of ascent at (u, v) where dual function g is non-differentiable, so we take best iterate so far:
$$g((u^{(k)}, v^{(k)})_{\text{best}}) = \max_{i=0,\ldots k} g(u^{(i)}, v^{(i)})$$

- The subgradient makes an angle < 90 with all ascent directions at (u, v)

$$f(y) \leq f(x) + s^\top (y - x) \quad \forall y \quad \Rightarrow \quad 0 < s^\top (y - x) \quad \forall f(y) > f(x)$$

This implies that a small move from (u, v) in the direction of any subgradient at u, v decreases the distance to any maximizer of g. To see this, let $v_{k+1} = v_k + t_k s_k$. Then

$$\|v_{k+1} - v^*\|^2 = \|v_k - v^*\|^2 + t_k^2 \|s_k\|^2 + 2 t_k s_k^\top (v_k - v^*)$$

Since $g(v_k) \leq g(v^*)$, we have

$$\|v_{k+1} - v^*\| \leq \|v_k - v^*\| \quad \forall 0 < t_k < 2(g(v^*) - g(v_k))/\|s_k\|^2$$

Step size choices

- Fixed step sizes: $t_k = t$ all $k = 1, 2, 3, \ldots$
- Diminishing step sizes: choose to meet conditions

$$\sum_{k=1}^{\infty} t_k^2 < \infty, \quad \sum_{k=1}^{\infty} t_k = \infty,$$

  i.e., square summable but not summable

  Important that step sizes go to zero, but not too fast

Other options too, but important difference to gradient descent: step sizes are typically pre-specified, not adaptively computed

Dual decomposition

Consider

$$\min_x \sum_{i=1}^B f_i(x_i) \quad \text{subject to} \quad \sum_{i=1}^B h_{ij}(x_i) \le 0 \quad j = 1, \ldots, m$$

Here $x = (x_1, \ldots x_B) \in \mathbb{R}^n$ divides into B blocks of variables, with each $x_i \in \mathbb{R}^{n_i}$.
Simple but powerful observation, in calculation of (sub)gradient, is that the minimization decomposes into B separate problems:

$$x^+ \in \operatorname*{argmin}_x \sum_{i=1}^B (f_i(x_i) + u^\top h_i(x_i))$$
$$\iff x_i^+ \in \operatorname*{argmin}_{x_i} \; f_i(x_i) + u^T h_i(x_i), \quad i = 1, \ldots B$$

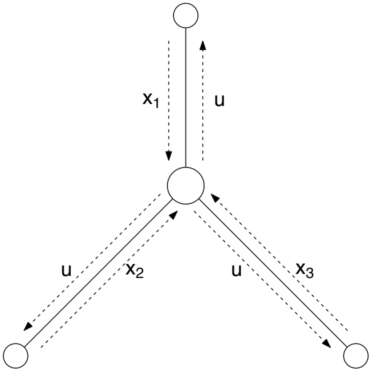
Dual decomposition algorithm: repeat for $k = 1, 2, 3, \dots$

$$x_i^{(k)} \in \underset{x_i}{\operatorname{argmin}} \; f_i(x_i) + (u^{(k-1)})^T h_i(x_i), \quad i = 1, \dots B$$

$$u^{(k)} = \max\left\{ u^{(k-1)} + t_k \left(\sum_{i=1}^{B} h_i(x_i^{(k)}) \right), 0 \right\}$$

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 $h_i(x_i)$ from each processor, update the global dual variable u



Price coordination interpretation (Vandenberghe):

- Have B units in a system, each unit chooses its own decision variable x_i (how to allocate its goods)
- There are m resources. Constraints are limits on shared resources ($\sum_{i=1}^{B} h_{ij}(x)$ is constraint on resource j), each component of dual variable u_j is price of resource j
- Dual update:

$$u_j^+ = (u_j + t\xi_j)_+, \quad j = 1, \ldots m$$

  where $\xi_j = \sum_{i=1}^{B} h_{ij}(x_i)$ are slacks
  - Increase price u_j if resource j is over-utilized, $\xi_j > 0$
  - Decrease price u_j if resource j is under-utilized, $\xi_j < 0$
  - Never let prices get negative