Dual Decomposition

Ryan Tibshirani
Convex Optimization 10-725
Last time: coordinate descent

Consider the problem

$$\min_x f(x)$$

where $$f(x) = g(x) + \sum_{i=1}^n h_i(x_i)$$, with $$g$$ convex and differentiable and each $$h_i$$ convex. **Coordinate descent**: let $$x^{(0)} \in \mathbb{R}^n$$, and repeat

$$x_i^{(k)} = \arg\min_{x_i} f(x_1^{(k)}, \ldots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, \ldots, x_n^{(k-1)})$$,

$$i = 1, \ldots, n$$

for $$k = 1, 2, 3, \ldots$$

- Very simple and easy to implement
- Careful implementations can achieve state-of-the-art
- Scalable, e.g., don’t need to keep full data in memory
Reminder: conjugate functions

Recall that given $f : \mathbb{R}^n \rightarrow \mathbb{R}$, 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,

$$x \in \partial f^*(y) \iff y \in \partial f(x) \iff x \in \arg\min_z f(z) - y^T z$$

- If $f$ is strictly convex, then $\nabla f^*(y) = \arg\min_z f(z) - y^T z$
Today:
- Dual ascent
- Dual decomposition
- Augmented Lagrangians
- A peak at ADMM
Dual first-order methods

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

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

\[ \partial g(u) = Ax - b \text{ where } x \in \arg\min_z f(z) + u^T Az \]

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

\[ x^{(k)} \in \arg\min_x f(x) + (u^{(k-1)})^T Ax \]

\[ u^{(k)} = u^{(k-1)} + t_k (Ax^{(k)} - b) \]

Step sizes \( t_k, k = 1, 2, 3, \ldots \), are chosen in standard ways.
Dual gradient ascent

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

$$x^{(k)} = \arg\min_x f(x) + (u^{(k-1)})^T Ax$$

$$u^{(k)} = u^{(k-1)} + t_k (Ax^{(k)} - b)$$

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

Lastly, proximal gradients and acceleration can be applied as they would usually
Curvature and conjugates

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

Proof of \(\Rightarrow\): Recall, if $g$ strongly convex with minimizer $x$, then

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

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

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

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

Adding these together, using Cauchy-Schwartz, rearranging shows that $\|x_u - x_v\|_2 \leq \|u - v\|_2/m$
Proof of “⇐”: for simplicity, call $g = f^*$ and $L = 1/m$. As $\nabla g$ is Lipschitz with constant $L$, so is $g_x(z) = g(z) - \nabla g(x)^T z$, hence

$$g_x(z) \leq g_x(y) + \nabla g_x(y)^T (z - y) + \frac{L}{2} \|z - y\|_2^2$$

Minimizing each side over $z$, and rearranging, gives

$$\frac{1}{2L} \|\nabla g(x) - \nabla g(y)\|_2^2 \leq g(y) - g(x) + \nabla g(x)^T (x - y)$$

Exchanging roles of $x, y$, and adding together, gives

$$\frac{1}{L} \|\nabla g(x) - \nabla g(y)\|_2^2 \leq (\nabla g(x) - \nabla g(y))^T (x - y)$$

Let $u = \nabla f(x)$, $v = \nabla g(y)$; then $x \in \partial g^*(u)$, $y \in \partial g^*(v)$, and the above reads $(x - y)^T (u - v) \geq \|u - v\|_2^2 / L$, implying the result
Convergence guarantees

The following results hold from combining the last fact with what we already know about gradient descent:\(^1\)

- If \( f \) is strongly convex with parameter \( m \), then dual gradient ascent with constant step sizes \( t_k = m \) converges at sublinear rate \( O(1/\epsilon) \)

- If \( f \) is strongly convex with parameter \( m \) and \( \nabla f \) is Lipschitz with parameter \( L \), then dual gradient ascent with step sizes \( t_k = 2/(1/m + 1/L) \) converges at linear rate \( O(\log(1/\epsilon)) \)

Note that this describes convergence in the dual. (Convergence in the primal requires more assumptions)

\(^1\)This is ignoring the role of \( A \), and thus reflects the case when the singular values of \( A \) are all close to 1. To be more precise, the step sizes here should be: \( m/\sigma_{\text{max}}(A)^2 \) (first case) and \( 2/(\sigma_{\text{max}}(A)^2/m + \sigma_{\text{min}}(A)^2/L) \) (second case).
Dual decomposition

Consider

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

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

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

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

$$x^+ \in \arg\min_x \sum_{i=1}^B f_i(x_i) + u^T Ax$$

$$\iff x_i^+ \in \arg\min_{x_i} f_i(x_i) + u^T A_i x_i, \quad i = 1, \ldots, B$$
Dual decomposition algorithm: repeat for \( k = 1, 2, 3, \ldots \)

\[
x_i^{(k)} \in \arg\min_{x_i} f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots, B
\]

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

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 \( u \)
Inequality constraints

Consider

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

Dual decomposition, i.e., projected subgradient method:

$$x_i^{(k)} \in \arg\min_{x_i} f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots, B$$

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

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

**Note:** The context suggests the use of slacks $s$ in the dual update formulation, which is a common approach in economic modeling to handle resource constraints.
Augmented Lagrangian method
(also known as: method of multipliers)

Dual ascent disadvantage: convergence requires strong conditions. Augmented Lagrangian method transforms the primal problem:

$$\begin{align*}
\min_x f(x) + \frac{\rho}{2} \|Ax - b\|_2^2 \\
\text{subject to } Ax = b
\end{align*}$$

where $\rho > 0$ is a parameter. Clearly equivalent to original problem. Strongly convex if $A$ has full column rank. Dual gradient ascent:

$$\begin{align*}
x^{(k)} &= \arg\min_x f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2} \|Ax - b\|_2^2 \\
u^{(k)} &= u^{(k-1)} + \rho (Ax^{(k)} - b)
\end{align*}$$
Notice step size choice \( t_k = \rho \) 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

\[
0 \in \partial f(x^{(k)}) + A^T \left( u^{(k-1)} + \rho(Ax^{(k)} - b) \right)
= \partial f(x^{(k)}) + A^T u^{(k)}
\]

This is the **stationarity condition** for original primal problem; under mild conditions \( Ax^{(k)} - b \to 0 \) as \( k \to \infty \), so KKT conditions are satisfied in the limit and \( x^{(k)}, u^{(k)} \) converge to solutions

- **Advantage:** augmented Lagrangian gives better convergence
- **Disadvantage:** lose decomposability! (Separability is ruined)
Alternating direction method of multipliers or ADMM: try for best of both worlds. Consider the problem

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

As before, we augment the objective

$$\min_x f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2$$

$$\text{subject to} \quad Ax + Bz = c$$

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 \( k = 1, 2, 3, \ldots \)

\[
x^{(k)} = \arg\min_x L_\rho(x, z^{(k-1)}, u^{(k-1)})
\]

\[
z^{(k)} = \arg\min_z L_\rho(x^{(k)}, z, u^{(k-1)})
\]

\[
u^{(k)} = u^{(k-1)} + \rho(Ax^{(k)} + Bz^{(k)} - c)
\]

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

\[
(x^{(k)}, z^{(k)}) = \arg\min_{x, z} L_\rho(x, z, u^{(k-1)})
\]
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**: $f(x^{(k)}) + g(z^{(k)}) \to f^* + g^*$, where $f^* + g^*$ is the optimal primal objective value

- **Dual convergence**: $u^{(k)} \to u^*$, where $u^*$ 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: roughly, ADMM behaves like first-order method. Theory still being developed, see, e.g., in Hong and Luo (2012), Deng and Yin (2012), Iutzeler et al. (2014), Nishihara et al. (2015)
Scaled form: denote $w = u/\rho$, so augmented Lagrangian becomes

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

and ADMM updates become

$$x^{(k)} = \arg\min_x f(x) + \frac{\rho}{2} \|Ax + Bz^{(k-1)} - c + w^{(k-1)}\|_2^2$$
$$z^{(k)} = \arg\min_z g(z) + \frac{\rho}{2} \|Ax^{(k)} + Bz - c + w^{(k-1)}\|_2^2$$
$$w^{(k)} = w^{(k-1)} + Ax^{(k)} + Bz^{(k)} - c$$

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

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

$$x^{(k)} = \arg\min_x P_C(z^{(k-1)} - w^{(k-1)})$$

$$z^{(k)} = \arg\min_z P_D(x^{(k)} + w^{(k-1)})$$

$$w^{(k)} = w^{(k-1)} + x^{(k)} - z^{(k)}$$
Compare classic alternating projections algorithm (von Neumann):

\[ x^{(k)} = \arg\min_x P_C(z^{(k-1)}) \]

\[ z^{(k)} = \arg\min_z P_D(x^{(k)}) \]

Difference is ADMM utilizes a dual variable \( w \) to offset projections. When (say) \( C \) is a linear subspace, ADMM algorithm becomes

\[ x^{(k)} = \arg\min_x P_C(z^{(k-1)}) \]

\[ z^{(k)} = \arg\min_z P_D(x^{(k)} + w^{(k-1)}) \]

\[ w^{(k)} = w^{(k-1)} + x^{(k)} - z^{(k)} \]

Initialized at \( z^{(0)} = y \), this is equivalent to Dykstra’s algorithm for finding the closest point in \( C \cap D \) to \( y \)
References


• L. Vandenberghe, Lecture Notes for EE 236C, UCLA, Spring 2011-2012