Dual methods and ADMM

Barnabas Poczos & Ryan Tibshirani
Convex Optimization 10-725/36-725

Recall conjugate functions

Given $f : \mathbb{R}^n \to \mathbb{R}$, the function

$$f^*(y) = \max_{x \in \mathbb{R}^n} \; y^T x - f(x)$$

is called its conjugate

- Conjugates appear frequently in dual programs, as

$$-f^*(y) = \min_{x \in \mathbb{R}^n} \; f(x) - y^T x$$

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

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

and for strictly convex f, $\nabla f^*(y) = \operatorname{argmin}_{z \in \mathbb{R}^n} (f(z) - y^T z)$

Outline

Today:

- Dual gradient methods
- Dual decomposition
- Augmented Lagrangians
- ADMM

Dual gradient methods

What if we can't derive dual (conjugate) in closed form, but want to utilize dual relationship? Turns out we can still use dual-based subradient or gradient methods

E.g., consider the problem

$$\min_{x \in \mathbb{R}^n} \; f(x) \;\; \text{subject to} \;\; Ax = b$$

Its dual problem is

$$\max_{u \in \mathbb{R}^m} \; -f^*(-A^T u) - b^T u$$

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

$$x \in \partial f^*(-A^T u) \quad \Longleftrightarrow \quad x \in \operatorname*{argmin}_{z \in \mathbb{R}^n} \; f(z) + u^T A z$$

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

$$
\begin{aligned}
x^{(k)} &\in \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) + (u^{(k-1)})^T A x \\
u^{(k)} &= u^{(k-1)} + t_k (A x^{(k-1)} - b)
\end{aligned}
$$

where t_k are step sizes, chosen in standard ways

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

$$
\begin{aligned}
x^{(k)} &= \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) + (u^{(k-1)})^T A x \\
u^{(k)} &= u^{(k-1)} + t_k (A x^{(k-1)} - b)
\end{aligned}
$$

(difference is that $x^{(k)}$ is unique, here)

Fact: if f strongly convex with parameter d, then ∇f^* Lipschitz with parameter $1/d$

Check: if f strongly convex and x is its minimizer, then

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

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:

$$d\|x_u - x_v\|_2^2 \leq (u - v)^T (x_u - x_v)$$

Using Cauchy-Schwartz, rearranging: $\|x_u - x_v\|_2 \leq (1/d) \cdot \|u - v\|_2$

Applying what we know about gradient descent: if f is strongly convex with parameter d, then dual gradient ascent with constant step size $t_k \leq d$ converges at rate $O(1/k)$. (Note: this is quite a strong assumption leading to a modest rate!)

Dual generalized gradient ascent and accelerated dual generalized gradient method carry through in similar manner

Disadvantages of dual methods:

- Can be slow to converge (think of subgradient method)
- Poor convergence properties: even though we may achieve convergence in dual objective value, convergence of $u^{(k)}, x^{(k)}$ to solutions requires strong assumptions (primal iterates $x^{(k)}$ can even end up being infeasible in limit)

Advantage: decomposability

Dual decomposition

Consider

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{B} f_i(x_i) \quad \text{subject to} \quad Ax = b$$

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

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

Simple but powerful observation, in calculation of (sub)gradient:

$$x^+ \in \operatorname*{argmin}_{x \in \mathbb{R}^n} \sum_{i=1}^{B} f_i(x_i) + u^T A x$$

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

i.e., minimization decomposes into B separate problems

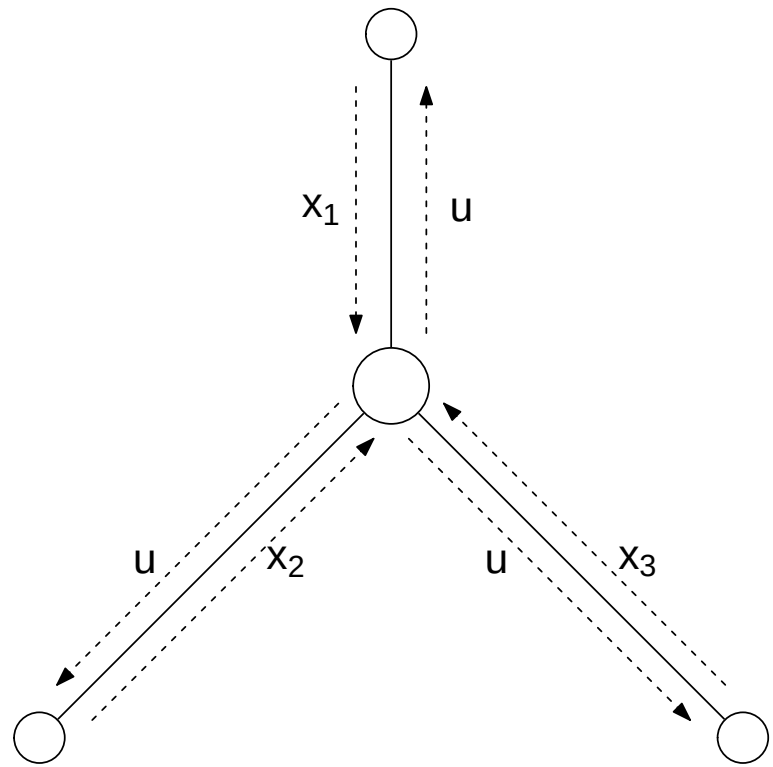
Dual decomposition algorithm: repeat for $k = 1, 2, 3, \ldots$

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

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

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



Example with inequality constraints:

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{B} f_i(x_i) \quad \text{subject to} \quad \sum_{i=1}^{B} A_i x_i \leq b$$

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

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

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

$$u^{(k)} = (v^{(k)})_+$$

where $(\cdot)_+$ is componentwise thresholding, $(u_+)_i = \max\{0, u_i\}$

Price coordination interpretation (from Vandenberghe's lecture notes):

- 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

Convergence of dual methods can be greatly improved by utilizing augmented Lagrangian. Start by transforming primal

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \; f(x) + \frac{\rho}{2}\|Ax - b\|_2^2 \\ &\text{subject to } Ax = b \end{aligned}$$

Clearly extra term $(\rho/2) \cdot \|Ax - b\|_2^2$ does not change problem

Assuming, e.g., A has full column rank, primal objective is strongly convex (parameter $\rho \cdot \sigma_{\min}^2(A)$), so dual objective is differentiable and we can use dual gradient ascent: repeat for $k = 1, 2, 3, \ldots$

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

Note step size choice $t_k = \rho$, for all k, in dual gradient ascent

Why? Since $x^{(k)}$ minimizes $f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2}\|Ax - b\|_2^2$ over $x \in \mathbb{R}^n$,

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

This is exactly the stationarity condition for the original primal problem; can show under mild conditions that $Ax^{(k)} - b$ approaches zero (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: not decomposable (separability compromised by augmented Lagrangian!)

ADMM

ADMM (Alternating Direction Method of Multipliers): go for the best of both worlds!

I.e., good convergence properties of augmented Lagrangians, along with decomposability

Consider minimization problem

$$\min_{x \in \mathbb{R}^n} \; f_1(x_1) + f_2(x_2) \;\; \text{subject to} \;\; A_1 x_1 + A_2 x_2 = b$$

As usual, we augment the objective

$$\min_{x \in \mathbb{R}^n} \; f_1(x_1) + f_2(x_2) + \frac{\rho}{2} \|A_1 x_1 + A_2 x_2 - b\|_2^2$$
$$\text{subject to } A_1 x_1 + A_2 x_2 = b$$

Write the augmented Lagrangian as

$$L_\rho(x_1, x_2, u) = f_1(x_1) + f_2(x_2) + u^T(A_1x_1 + A_2x_2 - b) + \frac{\rho}{2}\|A_1x_1 + A_2x_2 - b\|_2^2$$

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

$$\begin{aligned} x_1^{(k)} &= \operatorname*{argmin}_{x_1 \in \mathbb{R}^{n_1}} L_\rho(x_1, x_2^{(k-1)}, u^{(k-1)}) \\ x_2^{(k)} &= \operatorname*{argmin}_{x_2 \in \mathbb{R}^{n_2}} L_\rho(x_1^{(k)}, x_2, u^{(k-1)}) \\ u^{(k)} &= u^{(k-1)} + \rho(A_1x_1^{(k)} + A_2x_2^{(k)} - b) \end{aligned}$$

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

$$(x_1^{(k)}, x_2^{(k)}) = \operatorname*{argmin}_{(x_1, x_2) \in \mathbb{R}^n} L_\rho(x_1, x_2, u^{(k-1)})$$

Convergence guarantees

Under modest assumptions on f_1, f_2 (note: these do not require A_1, A_2 to be full rank), we get that ADMM iterates for any $\rho > 0$ satisfy:

- Residual convergence: $r^{(k)} = A_1 x_1^{(k)} - A_2 x_2^{(k)} - b \to 0$ as $k \to \infty$, i.e., primal iterates approach feasibility
- Objective convergence: $f_1(x_1^{(k)}) + f_2(x_2^{(k)}) \to f^\star$, where $f^\star$ is the optimal primal criterion 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 can be shown under more assumptions

Scaled form

It is often easier to express the ADMM algorithm in scaled form, where we replace the dual variable u by a scaled variable $w = u/\rho$

In this parametrization, the ADMM steps are

$$
\begin{aligned}
x_1^{(k)} &= \underset{x_1 \in \mathbb{R}^{n_1}}{\operatorname{argmin}} \; f_1(x_1) + \frac{\rho}{2}\|A_1 x_1 + A_2 x_2^{(k-1)} - b + w^{(k-1)}\|_2^2 \\
x_2^{(k)} &= \underset{x_2 \in \mathbb{R}^{n_2}}{\operatorname{argmin}} \; f_2(x_2) + \frac{\rho}{2}\|A_1 x_1^{(k)} + A_2 x_2 - b + w^{(k-1)}\|_2^2 \\
w^{(k)} &= w^{(k-1)} + A_1 x_1^{(k)} + A_2 x_2^{(k)} - b
\end{aligned}
$$

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

$$
w^{(k)} = w^{(0)} + \sum_{i=1}^{k} \left(A_1 x_1^{(i)} + A_2 x_2^{(i)} - b\right)
$$

Practicalities and tricks

In practice, ADMM obtains a relatively accurate solution in a handful of iterations, but requires many, many iterations for a highly accurate solution. Hence it behaves more like a first-order method than a second-order method

Choice of ρ can greatly influence practical convergence of ADMM

- ρ too large $\rightarrow$ not enough emphasis on minimizing $f_1 + f_2$
- ρ too small $\rightarrow$ not enough emphasis on feasibility

Boyd et al. (2010) give a strategy for varying ρ that is useful in practice (but without convergence guarantees)

Like deriving duals, getting a problem into ADMM form often requires a bit of trickery (and different forms can lead to different algorithms)

Alternating projections, revisited

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

$$\min_{x \in \mathbb{R}^n} \; 1_C(x) + 1_D(x)$$

To get into ADMM form, we write this as

$$\min_{x,z \in \mathbb{R}^n} \; 1_C(x) + 1_D(z) \;\; \text{subject to} \;\; x - z = 0$$

Each ADMM cycle involves two projections:

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \; P_C\big(z^{(k-1)} - w^{(k-1)}\big) \\ z^{(k)} &= \operatorname*{argmin}_{z \in \mathbb{R}^n} \; P_D\big(x^{(k)} + w^{(k-1)}\big) \\ w^{(k)} &= w^{(k-1)} + x^{(k)} - z^{(k)} \end{aligned}$$

This is like the classical alternating projections method, but now with a dual variable w (much more efficient)

Generalized lasso, revisited

Given the usual $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times p}$, and an additional $D \in \mathbb{R}^{m \times p}$, the generalized lasso problem solves

$$\min_{\beta \in \mathbb{R}^p} \ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \|D\beta\|_1$$

The generalized lasso is computationally harder than the lasso ($D = I$); recall our previous discussion on algorithms. Rewrite as

$$\min_{\beta \in \mathbb{R}^p,\, \alpha \in \mathbb{R}^m} \ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda \|\alpha\|_1 \ \text{ subject to } \ D\beta - \alpha = 0$$

and ADMM delivers a simple algorithm for the generalized lasso,

$$\begin{aligned}
\beta^{(k)} &= (X^T X + \rho D^T D)^+ \big(X^T y + \rho D^T (\alpha^{(k-1)} - w^{(k-1)})\big) \\
\alpha^{(k)} &= S_{\lambda/\rho}(D\beta^{(k)} + w^{(k-1)}) \\
w^{(k)} &= w^{(k-1)} + D\beta^{(k)} - \alpha^{(k)}
\end{aligned}$$

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