1 Outline

In this lecture, we study

- Dual gradient ascent,
- Proximal point algorithm,

2 Dual gradient ascent

We consider

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & Ax = b. \end{aligned}$$

We observed that its dual is given by

$$\text{maximize} \quad -f^*(-A^\top \mu) - b^\top \mu.$$

As f^* is convex, the dual problem is a concave maximization problem. Let us apply the gradient ascent method to the dual.

2.1 Superdifferential and the supergradient method

Definition 21.1. Given a concave function $f' : \mathbb{R}^d \to \mathbb{R}$ and a point $x \in \text{dom}(f')$, the *superdifferential* of f' at x is defined as

$$\partial f'(x) = \left\{ g : \ f'(y) \leq f'(x) + g^\top (y - x) \quad \forall y \in \text{dom}(f') \right\}.$$

Here, any $g \in \partial f(x)$ is called a *supergradient* of f' at x.

Note that $-f'$ is convex if f' is concave and that the subdifferential of $-f'$, given by $\partial(-f'(x))$ at a point x, is precisely $-\partial f'(x) = \{-g : \ g \in \partial(-f'(x))\}$. Hence, g is a supergradient of a concave function f' at a point $x \in \text{dom}(f')$ if and only if $-g$ is a subgradient of $-f'$ at x.

Furthermore, maximizing a concave function f' is equivalent to minimizing $-f'$ that is convex. Given a point x_t, let g_t be a supergradient of f' at x_t. Then $-g_t$ is a subgradient of $-f'$ at x_t, and the subgradient method applies the following update rule.

$$x_{t+1} = x_t - \eta_t(-g_t) = x_t + \eta_t g_t$$

for some step size $\eta_t > 0$. The algorithm that proceeds with this update rule is referred to as the supergradient method.

2.2 Supergradient method for the dual problem

Given μ_t, let $g_t \in \partial\left(-f^*(-A^\top \mu_t) - b^\top \mu_t\right)$. Then the supergradient method applies the following update rule.

$$\mu_{t+1} = \mu_t + \eta_t g_t.$$

Here, what is a supergradient g_t? Note that

$$\begin{aligned}
\underbrace{\partial\left(-f^*(-A^\top \mu_t) - b^\top \mu_t\right)}_{\text{superdifferential of } -f^*(-A^\top\mu)-b^\top\mu \text{ at } \mu=\mu_t} &= - \underbrace{\partial\left(f^*(-A^\top \mu_t) + b^\top \mu_t\right)}_{\text{subdifferential of } f^*(-A^\top\mu)+b^\top\mu \text{ at } \mu=\mu_t} \\
&= -\left(-A \underbrace{\partial f^*(-A^\top \mu_t)}_{\text{subdifferential of } f^*(\mu) \text{ at } \mu=-A^\top\mu_t} + b\right) \\
&= A\partial f^*(-A^\top \mu_t) - b.
\end{aligned}$$

Hence, $g_t \in \partial\left(-f^*(-A^\top \mu_t) - b^\top \mu_t\right)$ if and only if

$$g_t \in A\partial f^*(-A^\top \mu_t) - b.$$

Therefore,

$$g_t = Ax_t - b \quad \text{for some } x_t \in \partial f^*(-A^\top \mu_t).$$

Moreover, we have also observed that $x_t \in \partial f^*(-A^\top \mu_t)$ if and only if $-A^\top \mu_t \in \partial f(x_t)$. Here, $-A^\top \mu_t \in \partial f(x_t)$ holds if and only if $0 \in \partial f(x_t) + A^\top \mu_t$ which is equivalent to

$$x_t \in \operatorname*{argmin}_x f(x) + \mu_t^\top Ax.$$

Note that $\mu_t^\top b$ remains constant as x changes, so $x_t \in \operatorname{argmin}_x f(x) + \mu_t^\top Ax$ is equivalent to

$$x_t \in \operatorname*{argmin}_x f(x) + \mu_t^\top (Ax - b).$$

Therefore, the supergradient method applied to the dual problem proceeds with

$$\begin{aligned}
x_t &\in \operatorname*{argmin}_x f(x) + \mu_t^\top (Ax - b), \\
\mu_{t+1} &= \mu_t + \eta_t(Ax_t - b).
\end{aligned}$$

Here, $f(x) + \mu_t^\top (Ax - b)$ is the Lagrangian function $\mathcal{L}(x, \mu)$ at $\mu = \mu_t$. In words, the supergradient method applied to the dual problem works as follows. At each iteration t with a given dual multiplier μ_t, we find a minimizer of the Lagrangian function $\mathcal{L}(x, \mu_t)$. Then we use the corresponding dual supergradient $Ax_t - b$ to obtain a new multiplier μ_{t+1}.

Algorithm 1 Supergradient method for the dual problem

Initialize μ_1.
for $t = 1, \ldots, T-1$ **do**
  Obtain $x_t \in \operatorname{argmin}_x f(x) + \mu_t^\top (Ax - b)$,
  Update $\mu_{t+1} = \mu_t + \eta_t(Ax_t - b)$ for a step size $\eta_t > 0$.
end for

At each iteration, we find a minimizer of the Lagrangian function $\mathcal{L}(x, \mu_t)$, which gives rise to an unconstrained optimization problem. Hence, the dual approach is useful when there is a complex system of constraints.

2.3 Smoothness and strong convexity

Another motivation for using dual methods is that the dual objective can become smooth even if the primal objective is not.

Theorem 21.2. *Let $f : \mathbb{R}^d :\to \mathbb{R}$ be closed and α-strongly convex in the ℓ_2 norm. Then f^* is $(1/\alpha)$-smooth in the ℓ_2 norm.*

Proof. Given $y \in \mathbb{R}^d$, we have

$$f^*(y) = \sup_{x \in \text{dom}(f)} \left\{ y^\top x - f(x) \right\}.$$

Note that

$$\begin{aligned} x^* \in \partial f^*(y) \quad &\leftrightarrow \quad y \in \partial f(x^*) \\ &\leftrightarrow \quad 0 \in y - \partial f(x^*) \\ &\leftrightarrow \quad x^* \in \operatorname*{argmax}_{x \in \text{dom}(f)} \left\{ y^\top x - f(x) \right\}. \end{aligned}$$

Since f is strongly convex, there exists a unique maximizer x^* for the supremum. This implies that the subdifferential of f^* contains a unique point, and therefore, f^* is differentiable.

Let $y_1 \in \partial f(x_1)$ and $y_2 \in \partial f(x_2)$. Since f is α-strongly convex, we have

$$\begin{aligned} f(x_1) &\geq f(x_2) + y_2^\top (x_1 - x_2) + \frac{\alpha}{2} \|x_1 - x_2\|_2^2, \\ f(x_2) &\geq f(x_1) + y_1^\top (x_2 - x_1) + \frac{\alpha}{2} \|x_2 - x_1\|_2^2. \end{aligned}$$

Summing up these two inequalities, we obtain

$$(y_1 - y_2)^\top (x_1 - x_2) \geq \alpha \|x_1 - x_2\|_2^2.$$

Hence,

$$\|x_1 - x_2\|_2 \leq \frac{1}{\alpha} \|y_1 - y_2\|_2.$$

As $y_1 \in \partial f(x_1)$ and $y_2 \in \partial f(x_2)$, it follows that $x_1 = \nabla f^*(y_1)$ and $x_2 = \nabla f^*(y_2)$. Therefore,

$$\|\nabla f^*(y_1) - \nabla f^*(y_2)\|_2 \leq \frac{1}{\alpha} \|y_1 - y_2\|_2,$$

which implies that f^* is $(1/\alpha)$-smooth in the ℓ_2 norm. □

Remember that the subgradient method for strongly convex functions guarantees a convergence rate of $O(1/T)$. However, the dual problem of a strongly convex function minimization is a smooth convex function minimization, for which the accelerated gradient method guarantees a convergence rate of $O(1/T^2)$.

Theorem 21.3. *Let $f : \mathbb{R}^d :\to \mathbb{R}$ be a closed convex β-smooth function in the ℓ_2 norm. Then f^* is $(1/\beta)$-strongly convex in the ℓ_2 norm.*

Proof. To show that f^* is $(1/\beta)$-strongly convex in the ℓ_2 norm, we will argue that

$$h(y) = f^*(y) - \frac{1}{2\beta}\|y\|_2^2$$

is convex. Note that

$$\partial h(y) = \partial f^*(y) - \frac{1}{\beta}y.$$

We will use the fact that if ∂h is monotone, then h is convex. In other words, it is sufficient to show that for any $x_1 \in \partial f^*(y_1)$ and $x_2 \in \partial f^*(y_2)$, the following holds.

$$(y_1 - y_2)^\top ((x_1 - (1/\beta)y_1) - (x_2 - (1/\beta)y_2)) \geq 0,$$

which is equivalent to

$$(y_1 - y_2)^\top (x_1 - x_2) \geq \frac{1}{\beta}\|y_1 - y_2\|_2^2.$$

Remember that if f is β-smooth,

$$(\nabla f(x_1) - \nabla f(x_2))^\top (x_1 - x_2) \geq \frac{1}{\beta}\|\nabla f(x_1) - \nabla f(x_2)\|_2^2.$$

Moreover, for any $x_1 \in \partial f^*(y_1)$ and $x_2 \in \partial f^*(y_2)$, we have $y_1 = \nabla f(x_1)$ and $y_2 = \nabla f(x_2)$. Then the above inequality can be rewritten as

$$(y_1 - y_2)^\top (x_1 - x_2) \geq \frac{1}{\beta}\|y_1 - y_2\|_2^2,$$

as required. □

2.4 Dual gradient ascent for separable problems

We can use dual methods when the objective is separable while there is a system of linking constraints. We consider

$$\begin{aligned} \text{minimize} \quad & f_1(x_1) + f_2(x_2) \\ \text{subject to} \quad & A_1x_1 + A_2x_2 = b. \end{aligned}$$

Let us derive its dual. The Lagrangian dual function is given by

$$\begin{aligned} &\inf_{x_1,x_2} \left\{ f_1(x_1) + f_2(x_2) + \mu^\top(A_1x_1 + A_2x_2 - b) \right\} \\ &= -b^\top\mu + \inf_{x_1}\left\{ f_1(x_1) + \mu^\top A_1x_1 \right\} + \inf_{x_2}\left\{ f_2(x_2) + \mu^\top A_2x_2 \right\} \\ &= -b^\top\mu - \sup_{x_1}\left\{ -f_1(x_1) + (-A_1^\top\mu)^\top x_1 \right\} - \sup_{x_2}\left\{ -f_2(x_2) + (-A_2^\top\mu)^\top x_2 \right\} \\ &= -b^\top\mu - f_1^*(-A_1^\top\mu) - f_2^*(-A_2^\top\mu). \end{aligned}$$

Therefore, the Lagrangian dual problem is given by

$$\text{maximize} \quad -f_1^*(-A_1^\top\mu) - f_2^*(-A_2^\top\mu) - b^\top\mu.$$

Given μ_t, let $g_t \in \partial\left(-f_1^*(-A_1^\top\mu_t) - f_2^*(-A_2^\top\mu_t) - b^\top\mu_t\right)$. We can argue that

$$\partial\left(-f_1^*(-A_1^\top\mu_t) - f_2^*(-A_2^\top\mu_t) - b^\top\mu_t\right) = A_1\partial f_1^*(-A_1^\top\mu_t) + A_2\partial f_2^*(-A_2^\top\mu_t) - b.$$

Note that $x_{1,t} \in \partial f_1^*(-A_1^\top \mu_t)$ if and only if $-A_1^\top \mu_t \in \partial f_1(x_{1,t})$. This is equvialent to $x_{1,t} \in \operatorname{argmin}_{x_1} \left\{ f_1(x_1) + \mu_t^\top A_1 x_1 \right\}$. Similarly, $x_{2,t} \in \partial f_2^*(-A_2^\top \mu_t)$ if and only if $x_{2,t} \in \operatorname{argmin}_{x_2} \left\{ f_2(x_2) + \mu_t^\top A_2 x_2 \right\}$. Therefore, the supergradient method applied to the dual problem proceeds with the following update rule.

$$\mu_{t+1} = \mu_t + \eta_t(A_1 x_{1,t} + A_2 x_{2,t} - b)$$

where

$$x_{1,t} \in \operatorname*{argmin}_{x_1} \left\{ f_1(x_1) + \mu_t^\top A_1 x_1 \right\},$$
$$x_{2,t} \in \operatorname*{argmin}_{x_2} \left\{ f_2(x_2) + \mu_t^\top A_2 x_2 \right\}.$$

Algorithm 2 Supergradient method for the dual problem of a separable minimization

Initialize μ_1.
for $t = 1, \ldots, T-1$ **do**
  Obtain $x_{1,t} \in \operatorname{argmin}_{x_1} \left\{ f_1(x_1) + \mu_t^\top A_1 x_1 \right\}$ and $x_{2,t} \in \operatorname{argmin}_{x_2} \left\{ f_2(x_2) + \mu_t^\top A_2 x_2 \right\}$.
  $\mu_{t+1} = \mu_t + \eta_t(A_1 x_{1,t} + A_2 x_{2,t} - b)$ for a step size $\eta_t > 0$.
end for

Here, at each iteration, computing the iterates $x_{1,t}$ and $x_{2,t}$ can be done in parallel. For the primal problem, the variables x_1 and x_2 are connected through the constraints $A_1 x_1 + A_2 x_2 = b$. However, for the dual method, we separate the variables and x_1 and x_2 by the Lagrangian multiplier.

3 Proximal point algorithm

Remember that the proximal gradient method works for the following composite minimization problem.

$$\text{minimize} \quad f(x) = g(x) + h(x).$$

The proximal gradient method proceeds with the update rule

$$x_{t+1} = \operatorname{prox}_{\eta h}(x_t - \eta \nabla g(x)).$$

In this section, we discuss the proximal point method, which is a special case of proximal gradient, and its application to the dual problem. Note that minimizing a closed convex function f can be written as a (trivial) composite minimization as follows.

$$\text{minimize} \quad f(x) = 0 + f(x).$$

Here, the first part is $g = 0$, which is trivially smooth, and the second part is $h = f$. Then the corresponding proximal gradient update is given by

$$x_{t+1} = \operatorname{prox}_{\eta f}(x_t).$$

The algorithm with this update rule is referred to as the proximal point method. As $g = 0$ is smooth, the proximal point algorithm converges with a rate of $O(1/T)$.

Algorithm 3 Proximal point algorithm

Initialize x_1.
for $t = 1, \ldots, T$ **do**
  Update $x_{t+1} = \text{prox}_{\eta f}(x_t)$.
end for
Return x_{T+1}.

3.1 Proximal point algorithm and gradient descent

Theoretically, we can use any function h_t to run the proximal point algorithm, even if the objective is not h_t, in which case, the update rule corresponds to

$$x_{t+1} = \text{prox}_{\eta h_t}(x_t).$$

Hence, at each time step t, we may use a different function h_t hypothetically. Let us consider the first-order approximation of the objective function f at $x = x_t$.

$$h_t(x) = f(x_t) + \nabla f(x_t)^\top (x - x_t).$$

We know that $f(x) \geq h_t(x)$ for all x by convexity. Then what is the proximal point update with h_t? Note that

$$\begin{aligned} \text{prox}_{\eta h_t}(x_t) &= \operatorname*{argmin}_u \left\{ f(x_t) + \nabla f(x_t)^\top (u - x_t) + \frac{1}{2\eta}\|u - x_t\|_2^2 \right\} \\ &= x_t - \eta \nabla f(x_t). \end{aligned}$$

Therefore, the proximal point algorithm with the first-order approximation of f is precisely gradient descent. Hence, one can interpret gradient descent as an instance of the proximal point algorithm.

Let us now compare the proximal point algorithm with the objective f and gradient descent.

Lemma 21.4. $\text{prox}_{\eta f}(x) = (I + \eta \partial f)^{-1}(x)$.

Proof. Let $u = \text{prox}_{\eta f}(x)$. Remember that $u = \text{prox}_{\eta f}(x)$ if and only if $x - u \in \eta \partial f(u)$. Note that $x - u \in \eta \partial f(u)$ is equvialent to $x \in (I + \eta \partial f)(u)$, which is equivalent to $u \in (I + \eta \partial f)^{-1}(x)$. In summary,

$$u = \text{prox}_{\eta f}(x) \quad \leftrightarrow \quad u \in (I + \eta \partial f)^{-1}(x).$$

Since u is unique, it follows that $u = (I + \eta \partial f)^{-1}(x)$. □

By this lemma, the proximal point update rule can be written as

$$x_{t+1} = \text{prox}_{\eta f}(x_t) = (I + \eta \partial f)^{-1}(x_t).$$

This is equivalent to $x_t = (I + \eta \partial f)(x_{t+1}) = x_{t+1} + \eta \nabla f(x_{t+1})$, which is

$$x_{t+1} = x_t - \eta \nabla f(x_{t+1}).$$

In contrast to gradient descent that proceeds with $x_{t+1} = x_t - \eta \nabla f(x_t)$, we use the gradient at x_{t+1}.