1 Outline

In this lecture, we study

- Proximal gradient applied to the dual,
- Alternating direction method of multipliers (ADMM),
- Introduction to Newton's method

2 Dual of composite minimization

We consider

$$\text{minimize} \quad f(x) + g(Ax),$$

which is equivalent to

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

Its dual can be derived as

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

The gradient ascent method applied to the dual is given by the following.

Algorithm 1 Dual gradient method for composite problems

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

Basically, at each iteration, we minimize the Lagrangian function at $\mu = \mu_t$:

$$f(x) + g(y) + \mu_t^\top (Ax - y).$$

Instead, the augmented Lagrangian method considers the augmented Lagrangian function given by

$$f(x) + g(y) + \mu_t^\top (Ax - y) + \frac{\eta}{2}\|Ax - y\|_2^2.$$

Here, μ_t changes over iterations while η remains constant.

Algorithm 2 Augmented Lagrangian method for composite problems

Initialize μ_1.
for $t = 1, \dots, T-1$ **do**
  Obtain $(x_t, y_t) \in \operatorname{argmin}_{(x,y)} f(x) + g(y) + \mu_t^\top (Ax - y) + \frac{\eta}{2}\|Ax - y\|_2^2$,
  Update $\mu_{t+1} = \mu_t + \eta(Ax_t - y_t)$.
end for

2.1 Proximal gradient applied to the dual

Next, we apply the proximal gradient method to the dual. Throughout this subsection, let us assume that f^* is differentiable. Again, the dual is given by

$$\text{miniimize} \quad f^*(-A^\top \mu) + g^*(\mu).$$

The proximal gradient method proceeds with

$$\mu_{t+1} = \operatorname{prox}_{\eta g^*}\left(\mu_t + \eta A \nabla f^*(-A^\top \mu_t)\right)$$

since the gradient of $h(\mu) = f^*(-A^\top \mu)$ is $\nabla h(\mu) = -A\nabla f^*(-A^\top \mu)$. Moreover, $x_t = \nabla f^*(-A^\top \mu_t)$ if and only if $-A^\top \mu_t \in \partial f(x_t)$ which is equivalent to $x_t \in \operatorname{argmin}_x f(x) + \mu_t^\top Ax$. Hence, the update rule is equivalent to

$$\begin{aligned} x_t &\in \operatorname*{argmin}_x f(x) + \mu_t^\top Ax, \\ \mu_{t+1} &= \operatorname{prox}_{\eta g^*}\left(\mu_t + \eta A x_t\right). \end{aligned}$$

Furthermore, by the Moreau decomposition theorem, it follows that

$$\mu_{t+1} = \mu_t + \eta A x_t - \eta \operatorname{prox}_{g/\eta}(\mu_t/\eta + Ax_t).$$

Here, $y_t = \operatorname{prox}_{g/\eta}(\mu_t/\eta + Ax_t)$ if and only if

$$\frac{\mu_t}{\eta} + Ax_t - y_t \in \frac{1}{\eta}\partial g(y_t)$$

which is equivalent to

$$y_t \in \operatorname*{argmin}_y \left\{ g(y) + \mu_t^\top (Ax_t - y) + \frac{\eta}{2}\|Ax_t - y\|_2^2 \right\}.$$

Therefore, the proximal gradient descent applied to the dual is given by the following pseudo-code.

Algorithm 3 Proximal gradient for composite problems

Initialize μ_1.
for $t = 1, \dots, T-1$ **do**
  Obtain $x_t \in \operatorname{argmin}_x f(x) + \mu_t^\top Ax$,
  Obtain $y_t \in \operatorname{argmin}_y \left\{ g(y) + \mu_t^\top (Ax_t - y) + \frac{\eta}{2}\|Ax_t - y\|_2^2 \right\}$,
  Update $\mu_{t+1} = \mu_t + \eta(Ax_t - y_t)$.
end for

Algorithm 4 Alternating direction method of multipliers

Initialize μ_1 and y_0.
for $t = 1, \ldots, T-1$ **do**
  Obtain $x_t \in \operatorname{argmin}_x \left\{ f(x) + g(y_{t-1}) + \mu_t^\top (Ax - y_{t-1}) + \frac{\eta}{2}\|Ax - y_{t-1}\|_2^2 \right\}$,
  Obtain $y_t \in \operatorname{argmin}_y \left\{ f(x_t) + g(y) + \mu_t^\top (Ax_t - y) + \frac{\eta}{2}\|Ax_t - y\|_2^2 \right\}$,
  Update $\mu_{t+1} = \mu_t + \eta(Ax_t - y_t)$.
end for

2.2 ADMM

Lastly, we discuss the alternating direction method of multipliers (ADMM). Its pseudo-code is given by the following.

ADMM is equivalent to the Douglas-Rachford splitting method applied to the dual problem.

3 Newton's method

The update rule of gradient descent is to find the minimizer of a quadratic approximation of a given objective f. More precisely,

$$x_{t+1} \in \operatorname*{argmin}_x \left\{ f(x_t) + \nabla f(x_t)^\top (x - x_t) + \frac{1}{2\eta_t}\|x - x_t\|_2^2 \right\}.$$

Here, $f(x_t) + \nabla f(x_t)^\top (x - x_t)$ is the first-order approximation of f around x_t, and by adding the proximity term corresponding to the step size η_t, the resulting function becomes a quadratic. When f is twice-differentiable, the second-order approximation around x_t is

$$f(x_t) + \nabla f(x_t)^\top (x - x_t) + \frac{1}{2}(x - x_t)^\top \nabla^2 f(x_t)(x - x_t).$$

Hence, this is perhaps a better approximation than the one obtained by adding the proximity term to the first-order approximation. The convexity of f implies that $\nabla^2 f(x)$ is positive seminidefinite. If f is strictly convex, then $\nabla^2 f(x)$ is positive definite and thus invertible. Throughout this section, we focus on the setting where f is twice-differentiable and strongly convex, in which case f is strictly convex as well.

Let x_{t+1} be defined as the minimizer of the second-order approximation. Then, by the optimality condition, we have

$$x_{t+1} = x_t - \nabla^2 f(x_t)^{-1} \nabla f(x_t).$$

The algorithm the runs with this update rule is Newton's method. The following two propositions provide some intuition behind the direction $-\nabla^2 f(x_t)^{-1} \nabla f(x_t)$.

Proposition 23.1. *Direction $d = -\nabla^2 f(x_t)^{-1} \nabla f(x_t)$ is a descent direction at $x = x_t$.*

Proof. Remember that direction d is a descent direction at x_t if and only if $\nabla f(x_t)^\top d < 0$. Note that

$$\nabla f(x_t)^\top d = -\nabla f(x_t) \nabla^2 f(x_t)^{-1} \nabla f(x_t)$$

which is strictly negative because the Hessian $\nabla^2 f(x_t)$ is positive definite. □

Moreover,

Proposition 23.2. *Direction $d = -\nabla^2 f(x_t)^{-1}\nabla f(x_t)$ is a (scaled) steepest direction with respect to the quadratic norm $\|\cdot\|_{\nabla^2 f(x_t)}$ defined as*

$$\|y\|_{\nabla^2 f(x_t)} = (y^\top \nabla^2 f(x_t) y)^{1/2}.$$

Proof. Direction y is the steepest direction if and only if

$$\begin{aligned} y &\in \operatorname{argmin}\left\{\nabla f(x_t)^\top y : \ \|y\|_{\nabla^2 f(x_t)} \le 1\right\} \\ &= \operatorname{argmin}\left\{\nabla f(x_t)^\top y : \ y^\top \nabla^2 f(x_t) y \le 1\right\}. \end{aligned}$$

The Lagrangian function is defined as

$$\nabla f(x_t)^\top y + \lambda(y^\top \nabla^2 f(x_t) y - 1).$$

Here, we can check that

$$y^* = \frac{1}{\nabla f(x_t)^\top \nabla^2 f(x_t)^{-1} \nabla f(x_t)} d \quad \text{and} \quad \lambda^* = \frac{1}{2}$$

satisfy the KKT conditions. Note that $\nabla^2 f(x_t)^{-1}$ is also positive definite beacuse $\nabla^2 f(x_t)$ is positive definite, and therefore, $\nabla f(x_t)^\top \nabla^2 f(x_t)^{-1} \nabla f(x_t)$ is strictly positive as long as $\nabla f(x_t) \neq 0$. Therefore, d is a scaled steepest direction. □

Another intuition about using the direction $d = -\nabla^2 f(x_t)^{-1}\nabla f(x_t)$ is about reducing the gradient. The optimality condition is $\nabla f(x^*) = 0$, so we want to find a direction d such that

$$\nabla f(x_t + d) \approx 0.$$

Note that

$$\nabla f(x_t) + \nabla^2 f(x_t) d$$

is the first-order Tayler approximation of $\nabla f(x_t + d)$. Here, $d = -\nabla^2 f(x_t)^{-1}\nabla f(x_t)$ what makes $\nabla f(x_t) + \nabla^2 f(x_t) d$ set to 0.

3.1 Gradient descent and Newton's method

Let us consider the following quadratic minimization problem.

$$\text{minimize} \quad f(u, v) = \frac{1}{2} \begin{bmatrix} u & v \end{bmatrix}^\top \begin{bmatrix} M & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{2}(Mu^2 + mv^2).$$

Here, the objective function f is M-smooth and m-strongly convex in the ℓ_2 norm. Therefore, gradient descent converges to an ϵ-optimal solution after $O((M/m)\log(1/\epsilon))$ iterations.

Let us simulate gradient descent on f. Note that $\nabla f(u, v) = \begin{bmatrix} Mu & mv \end{bmatrix}^\top$. Let $x_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$. Then, as f is M-smooth, gradient descent proceeds with

$$x_2 = x_1 - \frac{1}{M}\nabla f(x_1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \frac{1}{M} \begin{bmatrix} 0 \\ m \end{bmatrix} = \left(1 - \frac{m}{M}\right) \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Moreover,

$$x_3 = x_2 - \frac{1}{M}\nabla f(x_2) = \left(1 - \frac{m}{M}\right) \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \frac{1}{M}\left(1 - \frac{m}{M}\right) \begin{bmatrix} 0 \\ m \end{bmatrix} = \left(1 - \frac{m}{M}\right)^2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Hence, we can see that

$$x_{t+1} = \left(1 - \frac{m}{M}\right)^t \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

As the optimal solution is $\begin{bmatrix} 0 & 0 \end{bmatrix}^\top$, the convergence rate is indeed $O((M/m)\log(1/\epsilon))$.

What about Newton's method? Note that

$$\nabla^2 f(u, v) = \begin{bmatrix} M & 0 \\ 0 & m \end{bmatrix}.$$

Hence, Netwon's method proceeds with

$$x_2 = x_1 - \nabla^2 f(x_1)^{-1} \nabla f(x_1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 1/M & 0 \\ 0 & 1/m \end{bmatrix} \begin{bmatrix} 0 \\ m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Therefore, after one iteration, Newton's method converges to the optimal solution. In fact, we will see that for functions that are both smooth and strongly convex, the convergence rate of Newton's method is $O(\log\log(1/\epsilon))$, which is much faster than gradient descent. Hence, if one wants to achieve a high accuracy, Newton's method is perhaps a better choice than gradient descent.