These notes cover some ideas about unconstrained minimization on convex functions. The material is adapted from [1].

1 Unconstrained minimization

In a previous lecture, we have seen an analysis of the convergence of gradient descent, with an upper bound on the number of iterations that is quadratic in $1/\varepsilon$. In this lecture we will show, with some stronger assumptions, an upper bound on the number of iterations that is logarithmic in $1/\varepsilon$. First we begin with some definitions.

1.1 Hessians and convexity

The Hessian of a function is a matrix encoding all of its second partial derivatives. We will define it in three different ways, abusing notation slightly and representing all three interpretations of the Hessian using the notation $\mathbf{Hf}$.

The first interpretation is the simplest, and it applies when $V = \mathbb{R}^n$. Then the Hessian is simply the matrix of second partial derivatives of f at $\mathbf{x}$:

$$[\mathbf{Hf_x}]_{ij} = \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j}.$$

Since f is continuously differentiable, $\mathbf{Hf_x}$ is symmetric.

The second interpretation applies whenever V has a non-degenerate inner-product structure, so that the gradient of f is well defined.

Definition 1.1. If $(V, \|\cdot\|)$ is a normed vector space and $f : V \to \mathbb{R}$ is twice continuously differentiable, the *Hessian* of f at $x \in S$, written $\mathbf{Hf_x}$, is a linear transformation satisfying for all $\mathbf{y}$,

$$\nabla \mathbf{f_{x+y}} = \nabla \mathbf{f_x} + \mathbf{Hf_x}(\mathbf{y}) + g(\mathbf{y})$$

where the remainder $g(\mathbf{y})$ vanishes to first order at $\mathbf{0}$. In other words, the Hessian is the Jacobian of the gradient.

The final definition, and the most general, does not even require an inner product structure on V. It defines the Hessian as a bilinear function $V \times V \to \mathbb{R}$ constituting the second-order

term in a local Taylor expansion of f,

$$f(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + \mathbf{df}_{\mathbf{x}}(\mathbf{y}) + \frac{1}{2}\mathbf{Hf}_{\mathbf{x}}(\mathbf{y},\mathbf{y}) + r(\mathbf{y})$$

where the remainder term $r(\mathbf{y})$ vanishes to second order at $\mathbf{y} = \mathbf{0}$.

The Hessian describes the local curvature of a function of many variables.

Lemma 1.2. *If $K \subseteq V$ is a closed, convex subset of V, a twice-differentiable function $f : K \to \mathbb{R}$ is convex if and only if $\mathbf{Hf}_{\mathbf{x}}$ is positive semi-definite for all $\mathbf{x} \in K$.*

Proof. We prove one direction. Assume $\mathbf{Hf}_{\mathbf{x}}$ is positive semi-definite for all $\mathbf{x} \in K$. By Taylor's theorem we have that

$$f(\mathbf{y}) = f(\mathbf{x}) + \langle \nabla \mathbf{f}_{\mathbf{x}}, \mathbf{y}-\mathbf{x}\rangle + \frac{1}{2}\langle \mathbf{y}-\mathbf{x}, \mathbf{Hf}_{\mathbf{z}}(\mathbf{y}-\mathbf{x})\rangle$$

for some $\mathbf{z}$ on the closed line segment from $\mathbf{x}$ to $\mathbf{y}$. Since $\mathbf{Hf}_{\mathbf{x}}$ is positive semi-definite, the last term is non-negative, so this implies

$$f(\mathbf{y}) - f(\mathbf{x}) \geq \langle \nabla \mathbf{f}_{\mathbf{x}}, \mathbf{y}-\mathbf{x}\rangle = \mathbf{df}_{\mathbf{x}}(\mathbf{y}) - \mathbf{df}_{\mathbf{x}}(\mathbf{x}).$$

The other direction is left as an exercise. □

As a simple example, consider the quadratic function $f\colon \mathbb{R}^n \to \mathbb{R}$,

$$f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, P\mathbf{x}\rangle + \langle \mathbf{x}, \mathbf{q}\rangle + r, \tag{1.1}$$

where $P \in \mathbb{R}^{n\times n}$ is symmetric positive semi-definite, $\mathbf{q} \in \mathbb{R}^n$, and $r \in \mathbb{R}$. Its gradient is $\nabla \mathbf{f}_{\mathbf{x}} = P\mathbf{x} + \mathbf{q}$, so its Hessian is $\mathbf{Hf}_{\mathbf{x}} = P$ and is independent of $\mathbf{x}$. In other words, the second-order Taylor approximation of a quadratic function is the function itself. If P is positive definite (instead of positive semi-definite), the equation $\nabla \mathbf{f}_{\mathbf{x}} = P\mathbf{x} + \mathbf{q} = 0$, which is a necessary and sufficient condition for $\mathbf{x}$ to be a minimizer of f, admits a unique solution, $\mathbf{x}^\star = -P^{-1}\mathbf{q}$.

A special case of minimizing a quadratic is the least-squares problem, which appears very frequently in a variety of contexts,

$$\text{minimize} \quad \|A\mathbf{x} - \mathbf{b}\|_2^2 = \langle \mathbf{x}, A^\top A\mathbf{x}\rangle - 2\langle \mathbf{x}, A^\top \mathbf{b}\rangle + \|\mathbf{b}\|_2^2.$$

The optimality condition leads to the famous *normal equations*,

$$A^\top A\mathbf{x}^\star = A^\top \mathbf{b}.$$

1.2 Strong convexity and smoothness

As the name suggests, strong convexity is a stronger assumption than convexity. In particular, the assumption made on the Hessian is stronger.

Definition 1.3. If $K \subseteq V$ is a closed, convex subset of V, a twice continuously differentiable function $f\colon K \to \mathbb{R}$ is said to be *α-strongly convex* if the smallest eigenvalue $\lambda_{\min}$ of $\mathbf{Hf_x}$ is at least $\alpha > 0$ for all $\mathbf{x} \in K$. Additionally, f is said to *β-smooth* if the largest eigenvalue $\lambda_{\max}$ of $\mathbf{Hf_x}$ is no greater than β for all $\mathbf{x} \in K$. The *condition number* of f is given by $\kappa = \beta/\alpha \geq 1$.

If we allowed $\alpha = 0$, then this reduces to the condition of convexity, since for a symmetric matrix $\lambda_{\min} = 0$ is an equivalent condition to being positive semi-definite.

For the quadratic function example given in Eq. (1.1), the definitions show that $f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, P\mathbf{x}\rangle$ is α-strongly convex and β-smooth if the eigenvalues of P all lie in the interval $[\alpha, \beta]$. In this case, the definition of the condition number κ coincides with the usual one, i.e., the ratio $\lambda_{\max}/\lambda_{\min}$. If P was the identity matrix, $\kappa = 1$ and f is said to perfectly conditioned as the level sets of f are perfectly concentric. If P had a large condition number $\kappa \gg 1$, then the level sets will be elongated, and algorithms like gradient descent may take long to find the minimum. Thus, the condition number κ can be thought of as a measure of difficulty of the convex minimization problem.

An example of a function that is convex, but not strongly convex nor linear is $f(\mathbf{x}) = (\langle \mathbf{a}, \mathbf{x}\rangle)^+ = \max\{\langle \mathbf{a}, \mathbf{x}\rangle, 0\}$ for $\mathbf{a} \neq 0$. In particular, $\nabla \mathbf{f_x} = 0$ when $\langle \mathbf{a}, \mathbf{x}\rangle < 0$, $\nabla \mathbf{f_x} = \mathbf{a}$ when $\langle \mathbf{a}, \mathbf{x}\rangle > 0$.

The conditions of strong convexity and smoothness give us the following.

Corollary 1.4. *If $K \subseteq V$ is a closed, convex subset of V, a twice continuously differentiable function $f\colon K \to \mathbb{R}$ that is α-strongly convex and β-smooth satisfies the following inequalities for all $\mathbf{y} \in K$:*

1. $$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \nabla \mathbf{f_x}, \mathbf{y} - \mathbf{x}\rangle + \frac{1}{2}\alpha\|\mathbf{y} - \mathbf{x}\|^2$$
2. $$f(\mathbf{y}) \leq f(\mathbf{x}) + \langle \nabla \mathbf{f_x}, \mathbf{y} - \mathbf{x}\rangle + \frac{1}{2}\beta\|\mathbf{y} - \mathbf{x}\|^2$$

i.e., f has local quadratic approximations that respectively lower- and upper-bound the function.

1.3 Gradient descent for strongly convex, smooth functions

We will now provide analysis for a modified version of gradient descent to minimize a function f that is α-strongly convex and β-smooth with condition number $\kappa = \beta/\alpha$. As before, we

will assume there is some constant difference between the starting point and the optimal value, i.e., $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^\star) \leq D$ where $\mathbf{x}^{(0)}$ is the starting point.

The difference between vanilla gradient descent and the algorithm presented below is that we will take the locally optimal (or nearly-locally optimal) step at each iteration, in the following sense. At each iteration, the algorithm moves in the direction of $\nabla \mathbf{f}_\mathbf{x}$ until it reaches the point along the ray $\{\mathbf{x} - t\nabla \mathbf{f}_\mathbf{x} \,|\, t \geq 0\}$ that minimizes f. This one-dimensional minimization problem on the parameter t is called *line search*, although a more accurate name might be *ray search*.

Algorithm 1 Gradient descent with line search

repeat
  $\Delta \mathbf{x} \leftarrow -\nabla \mathbf{f}_\mathbf{x}$
  $t_{\text{opt}} \leftarrow \arg\min_{t \geq 0} f(\mathbf{x} + t\Delta \mathbf{x})$
  $\mathbf{x} \leftarrow \mathbf{x} + t_{\text{opt}} \Delta \mathbf{x}$
until $\|\nabla \mathbf{f}_\mathbf{x}\|^2 \leq 2\varepsilon\alpha$
return $\mathbf{x}$

The minimization problem that defines t_{opt} can be solved exactly in cases where there is an exact analytical expression for the minimizer. Otherwise the minimization can be done approximately, to very high precision, via binary search. The binary search procedure is based on the observation that if we define $\mathbf{y} = \nabla \mathbf{f}_\mathbf{x}$ then the function $g(t) = \langle \mathbf{y}, \nabla \mathbf{f}_{\mathbf{x}+t\mathbf{y}} \rangle$ only changes sign once as t varies over $[0, \infty)$, and the sign change occurs at $t = t_{\text{opt}}$.

To see why the stopping criterion makes sense, note that the right-hand side of the strong convexity inequality in Corollary 1.4 is minimized with respect to $\mathbf{y}$ for $\tilde{\mathbf{y}} = \mathbf{x} - \frac{1}{\alpha}\nabla \mathbf{f}_\mathbf{x}$, and so we have the following inequalities

$$\begin{aligned} f(\mathbf{y}) &\geq f(\mathbf{x}) + \langle \nabla \mathbf{f}_\mathbf{x}, \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2}\alpha \|\mathbf{y} - \mathbf{x}\|^2 \\ &\geq f(\mathbf{x}) + \langle \nabla \mathbf{f}_\mathbf{x}, \tilde{\mathbf{y}} - \mathbf{x} \rangle + \frac{1}{2}\alpha \|\tilde{\mathbf{y}} - \mathbf{x}\|^2 \\ &= f(\mathbf{x}) - \frac{1}{2\alpha}\|\nabla \mathbf{f}_\mathbf{x}\|^2. \end{aligned}$$

In particular, the inequality holds for $f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \frac{1}{2\alpha}\|\nabla \mathbf{f}_\mathbf{x}\|^2$, so $\|\nabla \mathbf{f}_\mathbf{x}\|^2 \leq 2\varepsilon\alpha$ implies $f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \varepsilon$, where ε is the desired level of accuracy.

To bound the number of iterations, we will show that the quantity $f(\mathbf{x}) - f(\mathbf{x}^\star)$ decreases by a prescribed multiplicative factor in each iteration. First, we will analyze the line search procedure. From the smoothness inequality in Corollary 1.4, by setting $\mathbf{y} = \mathbf{x} - t\nabla \mathbf{f}_\mathbf{x}$ we have that

$$f(\mathbf{x} - t\nabla \mathbf{f}_\mathbf{x}) \leq f(\mathbf{x}) - t\|\nabla \mathbf{f}_\mathbf{x}\|^2 + \frac{1}{2}\beta t^2 \|\nabla \mathbf{f}_\mathbf{x}\|^2.$$

The right-hand side of this inequality is a quadratic that is minimized with respect to t

for $\tilde{t} = \frac{1}{\beta}$ and has minimum value $f(\mathbf{x}) - \frac{1}{2\beta}\|\nabla \mathbf{f}_{\mathbf{x}}\|^2$. Since t_{opt} was selected to minimize $f(\mathbf{x} - t\nabla \mathbf{f}_{\mathbf{x}})$ over $t \geq 0$, we have the following inequality

$$f(\mathbf{x} - t_{\text{opt}}\nabla \mathbf{f}_{\mathbf{x}}) \leq f(\mathbf{x}) - \frac{1}{2\beta}\|\nabla \mathbf{f}_{\mathbf{x}}\|^2.$$

Subtracting $f(\mathbf{x}^\star)$ from both sides, we have

$$f(\mathbf{x} - t_{\text{opt}}\nabla \mathbf{f}_{\mathbf{x}}) - f(\mathbf{x}^\star) \leq f(\mathbf{x}) - f(\mathbf{x}^\star) - \frac{1}{2\beta}\|\nabla \mathbf{f}_{\mathbf{x}}\|^2,$$

but we have already seen from our discussion of the stopping criterion that $\|\nabla \mathbf{f}_{\mathbf{x}}\|^2 \geq 2\alpha(f(\mathbf{x}) - f(\mathbf{x}^\star))$, so it follows that

$$f(\mathbf{x} - t_{\text{opt}}\nabla \mathbf{f}_{\mathbf{x}}) - f(\mathbf{x}^\star) \leq f(\mathbf{x}) - f(\mathbf{x}^\star) - \frac{\alpha}{\beta}(f(\mathbf{x}) - f(\mathbf{x}^\star)) = \left(1 - \frac{1}{\kappa}\right)(f(\mathbf{x}) - f(\mathbf{x}^\star)).$$

In words, in each iteration, the difference between optimality $f(\mathbf{x}) - f(\mathbf{x}^\star)$ shrinks by at least $\left(1 - \frac{1}{\kappa}\right)$. Since $\kappa \geq 1$, this is a well-defined and non-negative factor, and we observe that for 'ill-conditioned' problems with large values of κ, the bound is not as tight. Applying the inequality recursively, we have that

$$f(\mathbf{x}^{(j)}) - f(\mathbf{x}^\star) \leq \left(1 - \frac{1}{\kappa}\right)^j (f(\mathbf{x}^{(0)}) - f(\mathbf{x}^\star))$$

where $\mathbf{x}^{(j)}$ is the jth iterate, and thus in no more than $\log_{1-\frac{1}{\kappa}}(\varepsilon/D)$ iterations we have ε-optimality in the sense that $f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \varepsilon$. The base of this logarithm is hard to parse, but we can bound it from above by a simpler expression, using the inequality $\ln(1-x) \leq -x$, or equivalently, $-1/\ln(1-x) \leq 1/x$,

$$\log_{1-\frac{1}{\kappa}}(\varepsilon/D) = \frac{\ln(\varepsilon/D)}{\ln\left(1 - \frac{1}{\kappa}\right)} = \frac{\ln(D/\varepsilon)}{-\ln\left(1 - \frac{1}{\kappa}\right)} \leq \kappa \ln\left(\frac{D}{\varepsilon}\right).$$

The key things to notice about this upper bound are that it is linear in the condition number and logarithmic in $1/\varepsilon$. In contrast, the gradient descent algorithm from a previous lecture had a number of iterations that was quadratic in $1/\varepsilon$. Thus, the method is very fast when the Hessian of the convex function is not too ill-conditioned; for example when κ is is a constant, the number of iterations is merely logarithmic in $1/\varepsilon$.

Another thing to point out is that our bound on the number of iterations has *no dependence on the dimension*. Thus, the method is suitable even for very high-dimensional problems, as long as the high dimensionality does not lead to an excessively large condition number.

1.4 Newton's method

Newton's method is a powerful approach for unconstrained minimization, where we select our descent direction as the Newton's step, $\Delta \mathbf{x}_{\text{nt}} = -\mathbf{Hf}_{\mathbf{x}}{}^{-1}(\nabla \mathbf{f}_{\mathbf{x}})$.

Newton's method can be derived in many ways. One such is the minimization of the second-order Taylor approximation. The second-order approximation $\hat{f}$ of f at $\mathbf{x}$ is written

$$\hat{f}(\mathbf{x}+\mathbf{y}) = f(\mathbf{x}) + \langle \nabla \mathbf{f}_{\mathbf{x}}, \mathbf{y} \rangle + \frac{1}{2}\langle \mathbf{y}, \mathbf{Hf}_{\mathbf{x}}(\mathbf{y}) \rangle,$$

which is a convex quadratic function of $\mathbf{y}$, and is minimized when $\mathbf{y} = \Delta \mathbf{x}_{\text{nt}}$. The insight is that if the function f is quadratic, then $\mathbf{x} + \Delta \mathbf{x}_{\text{nt}}$ is the exact minimizer of f. However, even if the function f is only approximately quadratic, $\mathbf{x} + \Delta \mathbf{x}_{\text{nt}}$ is intuitively a good approximation of the minimizer of f.

Another interpretation of the Newton's step is as the solution to the linearized optimality condition $\nabla \mathbf{f}_{\mathbf{x}^\star} = 0$. For small $\mathbf{y}$ we have

$$\nabla \mathbf{f}_{\mathbf{x}+\mathbf{y}} \approx \nabla \mathbf{f}_{\mathbf{x}} + \mathbf{Hf}_{\mathbf{x}}(\mathbf{y}) = 0$$

which is a set of linear equations in $\mathbf{y}$, with solution $\mathbf{y} = \Delta \mathbf{x}_{\text{nt}}$. So the Newton step is what must be added to the current $\mathbf{x}$ for the linearized optimality condition to hold. This suggests that when $\mathbf{x}$ is near $\mathbf{x}^\star$, the update $\mathbf{x} + \Delta \mathbf{x}_{\text{nt}}$ will be a good approximation of $\mathbf{x}^\star$.

An important aspect of the Newton step is that it is equivariant to linear changes in coordinates. Suppose we have a change-of-basis matrix B, and define a new function related by this change of basis, $\tilde{f}(\mathbf{y}) = f(B\mathbf{y}) = f(\mathbf{x})$. Then

$$\nabla \tilde{\mathbf{f}}_{\mathbf{y}} = B^\top \nabla \mathbf{f}_{\mathbf{x}}, \qquad \mathbf{H}\tilde{\mathbf{f}}_{\mathbf{y}} = B^\top \mathbf{Hf}_{\mathbf{x}} B$$

and so the Newton step of $\tilde{f}$ at y is given by

$$\Delta \mathbf{y}_{\text{nt}} = -(B^\top \mathbf{Hf}_{\mathbf{x}} B)^{-1} B^\top \nabla \mathbf{f}_{\mathbf{x}} = B^{-1} \Delta \mathbf{x}_{\text{nt}}.$$

And we see that the Newton steps of f and $\tilde{f}$ are related by B, and further $\mathbf{x} + \Delta \mathbf{x}_{\text{nt}} = B(\mathbf{y} + \Delta \mathbf{y}_{\text{nt}})$. In contrast, we do not have a similar statement for gradient descent.

Newton's method also gives a better bound on the number of iterations, as compared to gradient descent with line search. We will not provide the details, but the bound looks like

$$\frac{D}{\gamma} + \log_2 \log_2(1/\varepsilon)$$

where as before, $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^\star) \leq D$ with $\mathbf{x}^{(0)}$ being the starting point. The value of $\gamma < 1$

depends on the parameters of an approximate line search, but is not too small. The analysis involves a 'damped' phase, in which the distance from optimum is reduced by a linear amount in each step, and a quadratically convergent phase, in which the distance from is reduced quadratically in each step. The value of $\log_2 \log_2(1/\varepsilon)$ is reasonably bounded above by the value 6, since machine-epsilon is usually 10^{-18}, and $\log_2 \log_2 10^{18} \approx 5.9$.

References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.