Introduction to Algorithms	Lecture Notes on Gradient Descent, Part 2
CS 4820, Spring 2014	Monday, May 5

Review

Recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if at each point $x \in \mathbb{R}^n$ there is a vector $\nabla f(x) \in \mathbb{R}^n$, called the *gradient* of f, such that the inequality

$$f(y) \geq f(x) + \nabla f(x)^\intercal (y - x)$$

is satisfied for all $y \in \mathbb{R}^n$. Vectors in $\mathbb{R}^n$ are interpreted as columns vectors, hence the notation $\nabla f(x)^\intercal (y - x)$ denotes the dot product of the vectors $\nabla f(x)$ and $y - x$. The linear function $\ell_x(y) = f(x) + \nabla f(x)^\intercal (y - x)$ can be interpreted as the function whose graph constitutes the tangent hyperplane to the graph of f at the point $(x, f(x))$.

In a *convex optimization* problem we are given the function f (represented in some form that allows us to evaluate f and its gradient) along with an initial point x_0. We are asked to find the minimizer x^* of f, i.e. the point $x^* \in \mathbb{R}^n$ where $f(x^*) = \min_{x \in \mathbb{R}^n} f(x)$. In reality, algorithms for convex minimization do not output the exact minimizer but merely a point at which the value of f is within ε of its global minimum value.

We make the following assumptions abaout f. (Throughout these notes, $\|y\|$ denotes the 2-norm of y, i.e. the Euclidean length of the vector y, which according to the Pythagorean Theorem is equal to square root of $y^\intercal y$.)

$$\|x^* - x_0\| \leq D \tag{1}$$

$$\|\nabla f(x)\| \leq L \qquad \forall x \in \mathbb{R}^n with \|x^* - x\| \leq D \tag{2}$$

Under those assumptions we saw that a gradient descent algorithm with fixed step size $\gamma = \varepsilon / L^2$ finds a point $\bar{x}$ at which $f(\bar{x}) \leq f(x^*) + \varepsilon$ in at most $T = L^2 D^2 / \varepsilon^2$ iterations.

Strongly convex functions

In this lecture our aim is to provide a gradient descent method that converges much more rapidly when the function f is *strongly convex*, which informally means that the curvature of f is not too close to zero. The material in this lecture is drawn from Boyd and Vandenberghe, *Convex Optimization*, published by Cambridge University Press and available for free download (with the publisher's permission) at `http://www.stanford.edu/~boyd/cvxbook/`.

We will make the following assumptions about f.

$$f(x_0) - f(x^*) \leq B \tag{3}$$

$$\frac{m}{2}\|y - x\|^2 \leq f(y) - \ell_x(y) \leq \frac{M}{2}\|y - x\|^2 \qquad \forall x, y \in \mathbb{R}^n \tag{4}$$

When f is twice differentiable, the second inequality is equivalent to requiring that the Hessian matrix $\nabla^2 f(x)$ has all of its eigenvalues between m and M, at every point $x \in \mathbb{R}^n$. The ratio M/m is thus an upper bound on the *condition number* of the Hessian of f. In geometric terms, when M/m is close to 1, it means that the level sets of f are nearly round, while if M/m is large it means that the level sets of f may be quite elongated.

We will analyze an algorithm which, in each iteration, moves in the direction of $-\nabla f(x)$ until it reaches the point on the ray $\{x - t\nabla f(x) \mid t \geq 0\}$ where the function f is (exactly or approximately) minimized. The advantage of this algorithm is that it is able to take large steps when the value of f is far from its minimum, and we will see that this is a tremendous advantage in terms of the number of iterations.

1: **repeat**
2: $\quad \Delta x = -\nabla f(x)$.
3: $\quad$ Choose $t \geq 0$ so as to minimize $f(x + t\Delta x)$.
4: $\quad x \leftarrow x + t\Delta x$.
5: **until** $\|\nabla f(x)\| \leq 2\varepsilon m$

To see why the stopping condition makes sense, observe that inequality (4) implies

$$\begin{aligned}
\ell_x(x^*) - f(x^*) &\leq -\frac{m}{2}\|x^* - x\|^2 \\
f(x) - f(x^*) &\leq \nabla f(x)^T(x^* - x) - \frac{m}{2}\|x^* - x\|^2 \\
f(x) - f(x^*) &\leq \min_{t\in\mathbb{R}} \|\nabla f(x)\|t - \frac{m}{2}t^2 \\
f(x) - f(x^*) &\leq \frac{\|\nabla f(x)\|^2}{2m}. \qquad (5)
\end{aligned}$$

The last line follows from basic calculus. The stopping condition $\|\nabla f(x)\|^2 \leq 2\varepsilon m$ ensures that $f(x) - f(x^*) \leq \varepsilon$ as desired.

To bound the number of iterations, we show that $f(x) - f(x^*)$ decreases by a prescribed multiplicative factor in each iteration. First observe that for any t,

$$\begin{aligned}
f(x + t\Delta x) - \ell_x(x + t\Delta x) &\leq \frac{M}{2}\|t\Delta x\|^2 = \frac{M}{2}\|\nabla f(x)\|^2 t^2 \\
f(x + t\Delta x) - f(x^*) &\leq \ell_x(x + t\Delta x) - f(x^*) + \frac{M}{2}\|\nabla f(x)\|^2 t^2 \\
&= f(x) - f(x^*) + \nabla f(x)^\intercal (t\Delta x) + \frac{M}{2}\|\nabla f(x)\|^2 t^2 \\
&\leq f(x) - f(x^*) - \|\nabla f(x)\|^2 t + \frac{M}{2}\|\nabla f(x)\|^2 t^2
\end{aligned}$$

The right side can be made as small as $f(x) - f(x^*) - \frac{\|\nabla f(x)\|^2}{2M}$ by setting $t = \frac{\|\nabla f(x)\|}{M}$. Our algorithm sets t to minimize the left side, hence

$$f(x + t\Delta x) - f(x^*) \leq f(x) - f(x^*) - \frac{\|\nabla f(x)\|^2}{2M}. \qquad (6)$$

Recalling from inequality (5) that $\|\nabla f(x)\|^2 \geq 2m(f(x) - f(x^*))$, we see that inequality (6) implies

$$f(x + t\Delta x) - f(x^*) \leq f(x) - f(x^*) - \frac{m}{M}[f(x) - f(x^*)] = \left(1 - \frac{m}{M}\right)[f(x) - f(x^*)]. \qquad (7)$$

This inequality shows that the difference $f(x) - f(x^*)$ shrinks by a factor of $1 - \frac{m}{M}$, or better, in each iteration. Thus, after no more than $\log_{1-m/M}(\varepsilon/B)$ iterations, we reach a point where $f(x) - f(x^*) \leq \varepsilon$, as was our goal. The expression $\log_{1-m/M}(\varepsilon/B)$ is somewhat hard to parse, but we can bound it from above by a simpler expression, by using the inequality $\ln(1 - x) \leq -x$.

$$\log_{1-m/M}(\varepsilon/B) = \frac{\ln(\varepsilon/B)}{\ln(1 - m/M)} = \frac{\ln(B/\varepsilon)}{-\ln(1 - m/M)} \leq \left(\frac{M}{m}\right)\ln\left(\frac{B}{\varepsilon}\right).$$

The key things to notice about this upper bound are that it is logarithmic in $1/\varepsilon$—as opposed to the algorithm from the previous lecture whose number of iterations was quadratic in $1/\varepsilon$—and that the number of iterations depends linearly on the condition number M/m. Thus, the method is very fast when the Hessian of the convex function is not too ill-conditioned; for example when M/m 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*, n. Thus, the method is suitable even for very high-dimensional problems, as long as the high dimensionality doesn't lead to an excessively large condition number M/m.