# 5 Nesterov's fast gradient method

In this lecture we will see that a simple (yet nontrivial!) modification of the gradient method allows us to boost the convergence rate from $O(1/k)$ to $O(1/k^2)$. The algorithm is as follows:

Start with $x_0 \in \mathbb{R}^n, \theta_0 = 1, v_0 = x_0$ and iterate for $k = 0, 1, \ldots$:

$$\begin{cases} \text{If } k \geq 1\text{: choose } \theta_k \in (0,1) \text{ so that } \frac{(1-\theta_k)t_k}{\theta_k^2} \leq \frac{t_{k-1}}{\theta_{k-1}^2} \\ y = (1-\theta_k)x_k + \theta_k v_k \\ x_{k+1} = y - t_k \nabla f(y) \\ v_{k+1} = x_k + \frac{1}{\theta_k}(x_{k+1} - x_k) \end{cases} \tag{1}$$

Some comments on the algorithm:

- The condition on $\theta_k$ looks complicated; it comes from the analysis of the sequences $\{x_k, v_k\}$. We will comment on the choice of $\theta_k$ later.
- The iterates $v_k$ can be eliminated. In this case, the algorithm has only two steps per iteration: $y = x_k + \beta_k(x_k - x_{k-1})$ where $\beta_k = \theta_k(\theta_{k-1}^{-1} - 1)$ and $x_{k+1} = y - t_k \nabla f(y)$. See Figure 5 for an illustration.
- Algorithm (1) is very similar to a standard gradient method: the "only" difference is that the gradient is taken at a point $y$ that is an extrapolation of $x_k$ along the direction $x_k - x_{k-1}$.
- The defining property of $v_{k+1}$ (last line of (1)) is that $x_{k+1} = (1-\theta_k)x_k + \theta_k v_{k+1}$. See also comment in Figure 5.

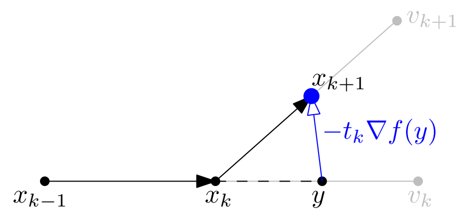


Figure 1: Iteration rule for the fast gradient method. $y$ is defined as an extrapolation of $x_k$ along the direction $x_k - x_{k-1}$, namely $y = x_k + \beta_k(x_k - x_{k-1})$. We evaluate the gradient of $f$ at $y$ and the new iterate is defined as $y - t_k \nabla f(y)$. We also show in this figure the iterates $v_k$. We show them in light gray because they are not "essential" for the algorithm (i.e., they can be eliminated). The only point to note here is that $y$ is a $\theta$-combination of $x_k$ and $v_k$; and $v_{k+1}$ is defined in such a way that $x_{k+1}$ is a $\theta$-combination (with the same $\theta$) of $x_k$ and $v_{k+1}$. It is easy to see from the picture that $v_{k+1} - v_k$ must be proportional to $\nabla f(y)$.

We now comment on the $\theta_k$'s:

- One can always find $\theta_k \in (0,1)$ such that the condition in the first line of the algorithm is always satisfied. In fact one can find a $\theta_k$ such that we have equality. This is given by $\theta_k = \frac{-a+\sqrt{a^2+4}}{2}$ where $a^2 = \theta_{k-1}^2 t_k / t_{k-1}$.

- When $t_k = t$ is fixed, one can check that the sequence $\theta_k = \frac{2}{k+2}$ satisfies the desired inequality $\frac{1-\theta_k}{\theta_k^2} \leq \frac{1}{\theta_{k-1}^2}$ (but it does not satisfy equality)

We are now ready to prove convergence of the algorithm:

**Theorem 5.1** (Nesterov)**.** *Let $f$ be convex with $L$-Lipschitz continuous gradient. The iterations of* (1) *with constant step size $t_k = t \in (0, 1/L]$ and with $\theta_k = \frac{2}{k+2}$ satisfy*

$$f(x_k) - f^* \leq \frac{2}{(k+1)^2 t}\|x_0 - x^*\|_2^2$$

*for all $k \geq 1$.*

*Proof.* We start like we did with the gradient method. We let $x^+ = y - t\nabla f(y)$. Then we have, since $0 < t \leq 1/L$

$$f(x^+) \leq f(y) + \nabla f(y)^T(x^+ - y) + \frac{1}{2t}\|x^+ - y\|_2^2. \tag{2}$$

By convexity of $f$ we also have, for any $z \in \mathbb{R}^n$, $f(y) - f(z) \leq \nabla f(y)^T(y - z)$. Combining this with (2), and using the fact that $\nabla f(y) = -\frac{1}{t}(x^+ - y)$ we get

$$f(x^+) \leq f(z) + \frac{1}{t}(x^+ - y)^T(z - x^+) + \frac{1}{2t}\|x^+ - y\|_2^2 \qquad \forall z \in \mathbb{R}^n. \tag{3}$$

Until now this is the same as for the analysis of the gradient method [In the gradient method we took $z = x^*$, rewrote the right hand side using completion of squares, and did the telescoping sum].

What we will do here is that we will evaluate (3) at $z = x^*$ and $z = x$ and consider the convex combination with weights $\{\theta, 1 - \theta\}$. This gives

$$\theta(f(x^+) - f(x^*)) + (1 - \theta)(f(x^+) - f(x)) \leq \frac{1}{t}(x^+ - y)^T(\theta x^* + (1 - \theta)x - x^+) + \frac{1}{2t}\|x^+ - y\|_2^2.$$

We simplify the above inequality as follows:

- The left-hand side can be written equivalently as $f(x^+) - f(x^*) - (1 - \theta)(f(x) - f(x^*))$
- The right-hand side is of the form $\frac{1}{2t}\left[2a^T b + \|a\|_2^2\right]$ (with $a = x^+ - y$ and $b = \theta x^* + (1-\theta)x - x^+$) which is equal to $\|a + b\|_2^2 - \|b\|_2^2$.

Thus we get

$$f(x^+) - f(x^*) - (1 - \theta)(f(x) - f(x^*)) \leq \frac{1}{2t}\left[\|\theta x^* + (1 - \theta)x - y\|_2^2 - \|\theta x^* + (1 - \theta)x - x^+\|_2^2\right]. \tag{4}$$

Now let's recall that $y = (1 - \theta)x + \theta v$ (where $v$ stands for $v_k$ and $v^+$ for $v_{k+1}$). This implies that the first-term on the RHS of (4) is $\theta^2\|x^* - v\|_2^2$. Also recall that $x^+ = (1 - \theta)x + \theta v^+$ and so the second-term on the RHS is (4) is $\theta^2\|x^* - v^+\|_2^2$. Finally we get

$$f(x_{k+1}) - f(x^*) - (1 - \theta_k)(f(x_k) - f(x^*)) \leq \frac{\theta_k^2}{2t}\left[\|x^* - v_k\|_2^2 - \|x^* - v_{k+1}\|_2^2\right] \tag{5}$$

Rearranging to put the iterates $k + 1$ on one side of the inequality, and the iterates $k$ on the other side:

$$\frac{t}{\theta_k^2}(f(x_{k+1}) - f(x^*)) + \frac{1}{2}\|x^* - v_{k+1}\|_2^2 \quad \leq \quad \frac{(1 - \theta_k)t}{\theta_k^2}(f(x_k) - f(x^*)) + \frac{1}{2}\|x^* - v_k\|_2^2 \tag{6}$$

Now we use the assumption that $(1-\theta_k)/\theta_k^2 \le 1/(\theta_{k-1})^2$ to get:

$$\frac{t}{\theta_k^2}(f(x_{k+1}) - f(x^*)) + \frac{1}{2}\|x^* - v_{k+1}\|_2^2 \quad \le \quad \frac{t}{\theta_{k-1}^2}(f(x_k) - f(x^*)) + \frac{1}{2}\|x^* - v_k\|_2^2. \tag{7}$$

Inequality above tells us that the quantity $V_k = \frac{t}{\theta_{k-1}^2}(f(x_k) - f(x^*)) + \frac{1}{2}\|x^* - v_k\|_2^2$ is nonincreasing with $k$. Thus we have $V_k \le V_{k-1} \le \cdots \le V_1$ which gives

$$\begin{aligned}
\frac{t}{\theta_{k-1}^2}(f(x_k) - f(x^*)) + \frac{1}{2}\|x^* - v_k\|_2^2 \quad &\le \quad \frac{t}{\theta_0^2}(f(x_1) - f(x^*)) + \frac{1}{2}\|x^* - v_1\|_2^2 \\
&\le \quad \frac{(1-\theta_0)t}{\theta_0^2}(f(x_0) - f(x^*)) + \frac{1}{2}\|x^* - v_0\|_2^2 \\
&= \quad \frac{1}{2}\|x^* - x_0\|_2^2
\end{aligned}$$

where the second line follows from (6) with $k = 0$, and the last line uses $\theta_0 = 1$ and $v_0 = x_0$. Thus we get $f(x_k) - f^* \le \frac{\theta_{k-1}^2}{2t}\|x^* - x_0\|_2^2$, and with $\theta_{k-1} = \frac{2}{k+1}$ we get the desired rate. □

Some remarks on the algorithm:

**Descent** The fast gradient method is not a descent method, i.e., it is possible that $f(x_{k+1}) > f(x_k)$ (unlike the gradient method). The convergence analysis proves however that a certain combination of $f(x_k) - f^*$ and $\|x^* - v_k\|_2^2$ decreases with $k$ (cf. Equation (7)).

**Backtracking line search** One can also prove convergence of the algorithm with a backtracking line search, rather than a constant line search. The only requirement on the step size $t_k$ is that inequality (2) is satisfied; this is the only thing needed in the convergence proof. The scheme works as follows: Starting with $t_k = \hat{t} > 0$, keep updating $t_k = \beta t_k$ with $\beta \in (0,1)$ until condition (2) is satisfied. (Note that the latter condition can be more succintly written as $f(x_{k+1}) \le f(y) - \frac{t_k}{2}\|\nabla f(y)\|_2^2$.) Also note that each time $t_k$ is updated, one has to recompute $\theta_k$, $y$, and $x_{k+1}$. In all, the line search at iteration $k$ proceeds as follows:

Start with $t_k = \hat{t}$, and compute associated $\theta_k, y, x_{k+1}$
While $f(x_{k+1}) > f(y) - \frac{t_k}{2}\|\nabla f(y)\|_2^2$
  Update $t_k = \beta t_k$
  Compute $\theta_k$ such that $\frac{1-\theta_k}{\theta_k^2} t_k \le \frac{t_{k-1}}{\theta_{k-1}^2}$
  Compute $y = (1-\theta_k)x_k + \theta_k v_k$
  Compute $x_{k+1} = y - t_k \nabla f(y)$

**Strongly convex case** We have seen in Lecture 3 that when the function $f$ is $m$-strongly convex, the gradient method with step size $t = 2/(m+L)$ converges at a linear rate $\approx (\frac{1-\kappa}{1+\kappa})^{2k}$ where $\kappa = \frac{m}{L}$. What about the fast gradient method? If we know the strong convexity parameter $m > 0$, algorithm (1) can be slightly modified to incorporate this knowledge. We do not give the general algorithm (as we did in Equation (1)), but only an important special case, where $t_k = 1/L$ and a specific choice of $\theta_k$. The algorithm reads:

$$\begin{cases} y = x_k + \frac{1-\sqrt{m/L}}{1+\sqrt{m/L}}(x_k - x_{k-1}) \\ x_{k+1} = y - (1/L)\nabla f(y). \end{cases} \tag{8}$$

One can prove that if $f$ is $m$-strongly convex and $\nabla f$ is $L$-Lipschitz, then the convergence rate of (8) is $\approx (1-\sqrt{\kappa})^{2k}$. This means that we reach accuracy $\epsilon$ in at most $O(\sqrt{\frac{L}{m}}\log(1/\epsilon))$ iterations. This can be much smaller than the $O(\frac{L}{m}\log(1/\epsilon))$ iterations of the gradient method [cf. Lecture 3].

One drawback of the algorithm (8) is that it relies on the knowledge of $m$ which can sometimes be difficult to estimate. (Note that the gradient method does not require knowledge of $m$. In lecture 3 we assumed $t_k = 2/(m+L)$ but one can easily see that $t_k = 1/L$ also gives a linear convergence rate of the form $(1-\kappa)^k$.) Several improvements and adaptations that avoid knowledge of $m$ have been proposed recently in the literature, see e.g., [OC15, Section 2.1].

*Lower bound for strongly convex functions:* One can prove that the convergence rate of $(1-\sqrt{\kappa})^{2k}$ achieved by (8) is the best possible for any first-order method that runs for $k$ iterations, for the class of functions $\mathcal{F}_{m,L} = \{f : \mathbb{R}^n \to \mathbb{R}\ m\text{-strongly convex and } L\text{-smooth}\}$. More precisely one can show (in a similar way as the proof in Lecture 4) that for any first-order algorithm $\mathcal{A}$ that runs for $k$ iterations, there is a function $f \in \mathcal{F}_{m,L}$ such that the $k$'th iterate of $\mathcal{A}$ on $f$ satisfies:

$$f(x_k) - f^* \gtrapprox m \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^{2k} \|x_0 - x^*\|^2.$$

# References

[OC15] Brendan O'Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015. 4