Accelerated first-order methods

We have seen three techniques for solving an unconstrained, smooth optimization program: gradient descent, Newton's method, and BFGS, which was an example of a quasi-Newton method. Gradient descent had the advantage of only needing to compute "first order" derivatives of the function we a minimizing (e.g. the gradient) — BFGS also has this property, but was implicitly trying to use changes in the first derivative to derive a second-order model.

There are small changes we can make to gradient descent that can dramatically improve its performance, both in theory and in practice. We talk about two of these here: the heavy ball method, and Nesterov's "optimal algorithm".

The Heavy Ball method

One way to interpret gradient descent is as a discretization to the *gradient flow* differential equation

$$\begin{aligned}\boldsymbol{x}'(t) &= -\nabla f(\boldsymbol{x}(t)),\\ \boldsymbol{x}(0) &= \boldsymbol{x}_0.\end{aligned}$$

The solution to these equations is a curve that tracks the direction of steepest descent directly to the minimizer. To see how gradient descent arises, we discretize the derivative with a forward difference

$$\boldsymbol{x}'(t) \approx \frac{\boldsymbol{x}(t+h) - \boldsymbol{x}}{h},$$

for some small h. So if we think of $\boldsymbol{x}^{(k+1)}$ and $\boldsymbol{x}^{(k)}$ as closely spaced time points, we can interpret

$$\frac{1}{\alpha}\left(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right) = -\nabla f(\boldsymbol{x}^{(k)}),$$

and a discrete approximation to gradient flow. Re-arranging the equation above yields the gradient descent iteration $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha\nabla f(\boldsymbol{x}^{(k)})$.

The problem is once we perform this discretization, the path tends to oscillate. One way to get a more regular path is to add a second-order term to the differential equation:

$$\boldsymbol{x}''(t) + a\boldsymbol{x}'(t) = -b\nabla f(\boldsymbol{x}(t))$$

From a physical perspective, this is a model for adding friction to a particle moving in potential field. Discretizing the dynamics as before with

$$\boldsymbol{x}''(t) \approx \frac{\boldsymbol{x}^{(k+1)} - 2\boldsymbol{x}^{(k)} + \boldsymbol{x}^{(k-1)}}{h_1}, \quad \boldsymbol{x}'(t) \approx \frac{\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}}{h_2}$$

gives us the iteration for the **heavy ball method**:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k\nabla f(\boldsymbol{x}^{(k)}) + \beta_k(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}).$$

The second term above adds a little bit of the last step $\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}$ directioninto the new step direction $\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ — this method is also referred to as *gradient descent with momentum.*

Convergence for strongly convex functions

We have seen that gradient descent exhibits linear convergence when $f(\boldsymbol{x})$ is strongly convex:

$$m\mathbf{I} \preceq \nabla^2 f(\boldsymbol{x}) \preceq M\mathbf{I}, \quad \text{for all } \boldsymbol{x} \in \operatorname{dom} f.$$

This means that the eigenvalues of the Hessian matrix are uniformly bounded between m and M at all points $\boldsymbol{x}$.

The analysis for gradient descent uses strong convexity with an application of the Taylor theorem. Briefly, we have

$$\begin{aligned}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^\star\|_2 &= \|\boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)}) - \boldsymbol{x}^\star\|_2 \\ &= \left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star - \alpha_k \left(\nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^\star)\right)\right\|_2 \\ &= \left\|\left(\mathbf{I} - \alpha_k \nabla^2 f(\boldsymbol{z})\right)\left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\right)\right\|_2 \\ &\leq \left\|\mathbf{I} - \alpha_k \nabla^2 f(\boldsymbol{z})\right\| \cdot \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2,\end{aligned}$$

where the second equality comes from the fact that $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$, while the third equality comes from the Taylor theorem: there exists some $\boldsymbol{z}$ on the line between $\boldsymbol{x}^{(k)}$ and $\boldsymbol{x}^\star$ such that

$$\nabla^2 f(\boldsymbol{z})(\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star) = \nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^\star).$$

Since we have a bound on the eigenvalues of $\nabla^2 f(\boldsymbol{z})$, we know that the maximum eigenvalue of the symmetric matrix $\mathbf{I} - \alpha_k \nabla^2 f(\boldsymbol{z})$ is no more than

$$\|\mathbf{I} - \alpha_k \nabla^2 f(\boldsymbol{z})\| \;\leq\; \max\left(|1 - \alpha_k m|, |1 - \alpha_k M|\right)$$

If we take $\alpha_k = 2/(M+m)$, we have

$$\|\mathbf{I} - \alpha_k \nabla^2 f(\boldsymbol{z})\| \;\leq\; \frac{M-m}{M+m} = \frac{\kappa - 1}{\kappa + 1},$$

where $\kappa = M/m$ is the "condition number". So we have

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^\star\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right) \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2,$$

which, by induction on k means

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2.$$

When κ is even moderately large, we have

$$\frac{\kappa-1}{\kappa+1} \approx 1 - \frac{1}{\kappa},$$

which means that in order to guarantee that we have reduced the initial error by a factor of ϵ, $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2 / \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2 \leq \epsilon$, we need

$$k \ \gtrsim\ \kappa \cdot \log(1/\epsilon)$$

For the heavy ball method, we have a similar analysis that ends in a better result. We start by looking at how $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^\star\|^2 + \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2^2$ goes to zero for fixed values of α, β which we will choose later:

$$\begin{aligned}
\left\| \begin{bmatrix} \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^\star \\ \boldsymbol{x}^{(k)} - \boldsymbol{x}^\star \end{bmatrix} \right\|_2 &= \left\| \begin{bmatrix} \boldsymbol{x}^{(k)} + \beta(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}) - \boldsymbol{x}^\star \\ \boldsymbol{x}^{(k)} - \boldsymbol{x}^\star \end{bmatrix} - \alpha \begin{bmatrix} \nabla f(\boldsymbol{x}^{(k)}) - \nabla f(\boldsymbol{x}^\star) \\ \mathbf{0} \end{bmatrix} \right\|_2 \\
&= \left\| \begin{bmatrix} \boldsymbol{x}^{(k)} + \beta(\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}) - \boldsymbol{x}^\star \\ \boldsymbol{x}^{(k)} - \boldsymbol{x}^\star \end{bmatrix} - \alpha \begin{bmatrix} \nabla^2 f(\boldsymbol{z}) \left(\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\right) \\ \mathbf{0} \end{bmatrix} \right\|_2 \\
&= \left\| \begin{bmatrix} (1+\beta)\mathbf{I} - \alpha\nabla^2 f(\boldsymbol{z}) & -\beta\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}^{(k)} - \boldsymbol{x}^\star \\ \boldsymbol{x}^{(k-1)} - \boldsymbol{x}^\star \end{bmatrix} \right\|_2 .
\end{aligned}$$

Applying the above iteratively means that, in the limit

$$\left\| \boldsymbol{x}^{(k)} - \boldsymbol{x}^\star \right\|_2 \ \lesssim\ \|\boldsymbol{T}^k\|\, \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2.$$

where

$$\boldsymbol{T} = \begin{bmatrix} (1+\beta)\mathbf{I} - \alpha\nabla^2 f(\boldsymbol{z}) & -\beta\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}.$$

It is a fundamental result from numerical linear algebra that as k gets large, $\|\boldsymbol{T}^k\|$ becomes very close to[1] $\rho(\boldsymbol{T})^k$, where ρ is the maximum of the magnitudes of the eigenvalues of $\boldsymbol{T}$.

[1] If $\boldsymbol{T}$ is symmetric, then of course $\|\boldsymbol{T}^k\| = \rho(\boldsymbol{T})^k$ for all k.

We can get at this eigenvalues in a systematic way. Start by taking an eigenvalue decomposition of the Hessian matrix above, $\nabla^2 f(\boldsymbol{z}) = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{\mathrm{T}}$. Since $\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \mathbf{I}$, we can write

$$\begin{bmatrix} (1+\beta)\mathbf{I} - \alpha\nabla^2 f(\boldsymbol{z}) & -\beta\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{V} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{V} \end{bmatrix} \begin{bmatrix} (1+\beta)\mathbf{I} - \alpha\boldsymbol{\Lambda} & -\beta\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{V}^{\mathrm{T}} \end{bmatrix}.$$

Since $\begin{bmatrix} \boldsymbol{V} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{V} \end{bmatrix}$ is orthonormal, its application on the left and the right of a matrix does not change the magnitude of its eigenvalues, and we have

$$\begin{aligned} \rho(\boldsymbol{T}) &= \rho\left(\begin{bmatrix} (1+\beta)\mathbf{I} - \alpha\boldsymbol{\Lambda} & -\beta\mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}\right) \\ &= \max_{n=1,\ldots,N} \rho(\boldsymbol{T}_n), \end{aligned}$$

where

$$\boldsymbol{T}_n = \begin{bmatrix} 1+\beta-\alpha\lambda_n & -\beta \\ 1 & 0 \end{bmatrix}.$$

This last equality follows from the fact the large $2N \times 2N$ matrix can have its rows and columns permuted (which doesn't change the eigenvalues) to become block diagonal, with the 2×2 matrices $\boldsymbol{T}_n$ as the blocks.

We now have the problem of finding α, β that minimize the size of the largest eigenvalue of the 2×2 matrices above given the knowledge that $m \leq \lambda_n \leq M$. A very technical calculation yields the fact that taking

$$\alpha = \frac{4}{(\sqrt{M} - \sqrt{m})^2}, \quad \beta = \frac{\sqrt{M} - \sqrt{m}}{\sqrt{M} + \sqrt{m}},$$

yields

$$\rho(\boldsymbol{T}_n) \leq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \quad \text{for all } n.$$

Thus the convergence of the heavy ball method is approximately

$$\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\right\|_2 \;\lesssim\; \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2$$

Again, for only moderately large κ, we have

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \approx 1 - \frac{1}{\sqrt{\kappa}},$$

meaning that

$$\frac{\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^\star\|_2}{\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2} \le \epsilon \quad \text{when} \quad k \gtrsim \sqrt{\kappa}\,\log(1/\epsilon).$$

The difference with gradient descent can be significant. When $\kappa = 10^2$, we are asking for $\approx 100\log(1/\epsilon)$ iterations for gradient descent, as compared with $\approx 10\log(1/\epsilon)$ from the heavy ball method.

Nesterov's "optimal" method

In the case where f is strictly convex, you can come up with examples that show that the convergence rate of the heavy ball method can't be improved in general. For non-strictly convex f, the story is more complicated.

Recall from a few lectures ago that we also had the convergence result for gradient descent in the case where $m = 0$; that is, we only know that the gradient is Lipschitz:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \;\leq\; M\|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

We saw that for gradient descent

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) \leq \text{Const}\,\frac{M}{k}\,\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2$$

where $p^\star$ is the minimal value of f. So to reduce the error by a factor of ϵ requires

$$k \gtrsim 1/\epsilon$$

iterations.

There is a trick similar to heavy ball, but with yet another extra term, that can improve on this in theory, and often works better in practice. We have the iteration

$$\begin{aligned}
&\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)} \\
&\boldsymbol{p}^{(k+1)} = -\nabla f\left(\boldsymbol{x}^{(k+1)} + \beta_{k+1}(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)})\right) + \beta_{k+1}\boldsymbol{p}^{(k)}.
\end{aligned}$$

(We start with $\boldsymbol{p}^{(0)} = \mathbf{0}$.) Notice that this is the same as heavy ball *except* that there is also a momentum term *inside* the gradient expression. With this iteration, it can be shown that

$$f(\boldsymbol{x}^{(k)}) - f(\boldsymbol{x}^\star) \leq \mathrm{Const}\,\frac{M}{k^2}\,\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|_2,$$

meaning that we can reduce the error by a factor of ϵ in

$$k \gtrsim 1/\sqrt{\epsilon},$$

iterations. When $\epsilon \sim 10^{-4}$, this is much, much better than $1/\epsilon$.

Nesterov's method is called "optimal" because it is impossible to beat the $1/k^2$ rate using only function and gradient evaluations. There are careful demonstrations of this in the literature.

The iteration above also tends to work better in practice. Nesterov's paper (which appeared in Russian in 1983, then in English in 2004) contains a sequence of subtle calculations that show why this method is better, yet it appears hard to abstract some higher-level understanding of what he did. I have heard the phrase "there is no intuition, purely calculation" on multiple occasions when people are talking about this method.

There are also good, practical ways that the stepsize α_k and the momentum β_k can be chosen. In practice, α_k can be chosen using a standard line search, and a good choice of β (both in theory and in practice) turns out to be

$$\beta_k = \frac{k-1}{k+2}.$$