Introduction to Optimization

Improving convergence rates.

Lecture 07: Quadratic functions and finite termination.

Lecture 08: Inertial algorithms. Stochastic gradient.

university of groningen

Momentum, inertia, acceleration

$x_{n+1} = x_n - \alpha_n \nabla f(x_n)$ is equivalent to

$$-\frac{x_{n+1} - x_n}{\alpha_n} = \nabla f(x_n),$$

which is an approximation of the steepest descent evolution equation

$$-\dot{x}(t) = \nabla f\big(x(t)\big).$$

Momentum, inertia, acceleration

$x_{n+1} = x_n - \alpha_n \nabla f(x_n)$ is equivalent to

$$-\frac{x_{n+1} - x_n}{\alpha_n} = \nabla f(x_n),$$

which is an approximation of the steepest descent evolution equation

$$-\dot{x}(t) = \nabla f\big(x(t)\big).$$

Other dynamics are related to minimization of potentials. For example,

$$m\ddot{x}(t) + \gamma \dot{x}(t) + \nabla f\big(x(t)\big) = 0.$$

Discretization

We discretize

$$m\ddot{x}(t) + \gamma\dot{x}(t) + \nabla f\big(x(t)\big) = 0$$

to obtain

$$m\,\frac{x_{n+1} - 2x_n + x_{n-1}}{h^2} + \gamma\,\frac{x_n - x_{n-1}}{h} + \nabla f(y_n) = 0.$$

Discretization

We discretize

$$m\ddot{x}(t) + \gamma\dot{x}(t) + \nabla f\big(x(t)\big) = 0$$

to obtain

$$m\,\frac{x_{n+1} - 2x_n + x_{n-1}}{h^2} + \gamma\,\frac{x_n - x_{n-1}}{h} + \nabla f(y_n) = 0.$$

Equivalently,

$$x_{n+1} = x_n + \beta_n\,(x_n - x_{n-1}) - \alpha_n\,\nabla f(y_n),$$

with $\alpha_n = \frac{h^2}{m}$ and $\beta_n = 1 - \frac{\gamma h}{m}$.

Two popular choices

Polyak's heavy ball (1964)

$$x_{n+1} = x_n + \beta_n (x_n - x_{n-1}) - \alpha_n \nabla f(x_n).$$

Two popular choices

Polyak's heavy ball (1964)

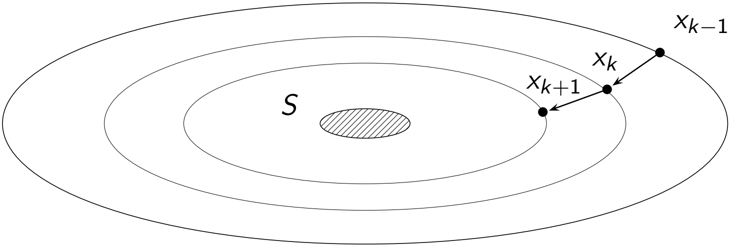
$$x_{n+1} = x_n + \beta_n (x_n - x_{n-1}) - \alpha_n \nabla f(x_n).$$

Nesterov's extrapolation (1983)

$$\begin{cases} y_n & = & x_n + \beta_n (x_n - x_{n-1}) \\ x_{n+1} & = & y_n - \alpha_n \nabla f(y_n). \end{cases}$$

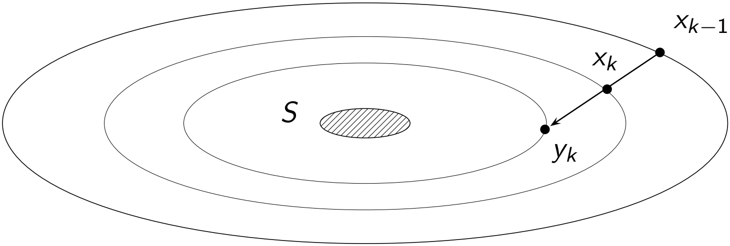
Nesterov's extrapolation

The main idea is the following: Instead of doing this



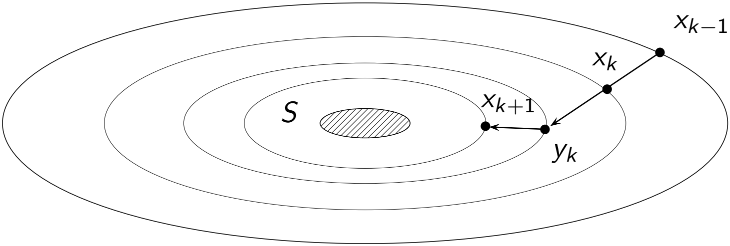
Nesterov's extrapolation

Better try this



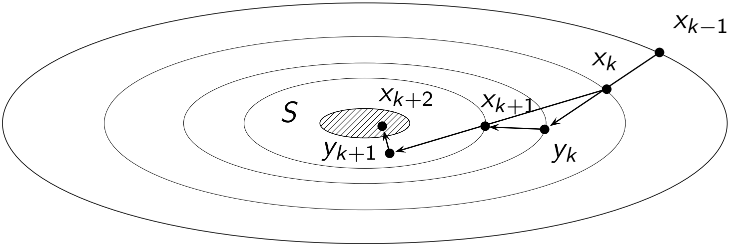
Nesterov's extrapolation

Better try this



Nesterov's extrapolation

Better try this



Convergence of Nesterov's method

Theorem

Let $f : \mathbb{R}^N \to \mathbb{R}$ *be an L-smooth convex function with minimizers, and let* (x_n, y_n) *be generated by Nesterov's method with convenient* α_n, β_n.

Convergence of Nesterov's method

Theorem

Let $f : \mathbb{R}^N \to \mathbb{R}$ be an L-smooth convex function with minimizers, and let (x_n, y_n) be generated by Nesterov's method with convenient α_n, β_n.

- *Then,* $f(x_n) - \min(f) \leq \dfrac{L\,\mathrm{dist}(x_0, S)^2}{(n+1)^2}$ *for all* $n \geq 1$*. In addition,* $\lim\limits_{n\to\infty} n^2\big(f(x_n) - \min(f)\big) = 0$*.*

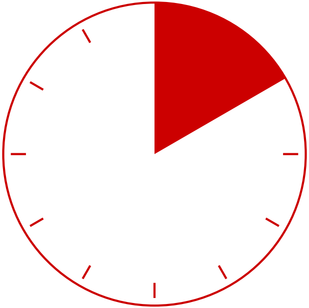
Convergence of Nesterov's method

Theorem

Let $f : \mathbb{R}^N \to \mathbb{R}$ be an L-smooth convex function with minimizers, and let (x_n, y_n) be generated by Nesterov's method with convenient α_n, β_n.

- *Then,* $f(x_n) - \min(f) \leq \dfrac{L\,\mathrm{dist}(x_0, S)^2}{(n+1)^2}$ *for all* $n \geq 1$. *In addition,* $\lim\limits_{n\to\infty} n^2\big(f(x_n) - \min(f)\big) = 0$.
- *If, moreover, f is* μ*-strongly convex, then*
$$f(x_n) - \min(f) \leq L\,\mathrm{dist}(x_0, S)^2 \left(1 - \sqrt{\frac{\mu}{L}}\right)^k \quad \text{for all } n \geq 1.$$

Break



Let $f : \mathbb{R}^N \to \mathbb{R}$, and let Ξ be a probability space.

Let $f : \mathbb{R}^N \to \mathbb{R}$, and let Ξ be a probability space.

The stochastic gradient method is defined by

$$x_{n+1} = x_n - \alpha_n \, g(x_n, \xi_n),$$

where $\alpha_n > 0$, (ξ_n) is an i.i.d. sequence of random variables in Ξ, and $g : \mathbb{R}^N \times \Xi$ is intended to approximate ∇f in the sense that

$$\mathbb{E}_\xi \big(g(x, \xi) \big) \sim \nabla f(x)$$

for all $x \in \mathbb{R}^N$.

Examples

1. Noisy Gradients: $g(x,\xi) = \nabla f(x) + \xi$, with $\mathbb{E}(\xi) = 0$.

Examples

1. Noisy Gradients: $g(x, \xi) = \nabla f(x) + \xi$, with $\mathbb{E}(\xi) = 0$.

2. Incremental Gradient Method: For $f = \dfrac{1}{M}\displaystyle\sum_{m=1}^{M} f_m$, at iteration n, we select $j_n \in \{1, \ldots, M\}$ and compute $x_{n+1} = x_n - \alpha_n \nabla f_{j_n}(x_n)$.

Examples

1. Noisy Gradients: $g(x,\xi) = \nabla f(x) + \xi$, with $\mathbb{E}(\xi) = 0$.

2. Incremental Gradient Method: For $f = \dfrac{1}{M}\displaystyle\sum_{m=1}^{M} f_m$, at iteration n, we select $j_n \in \{1,\ldots,M\}$ and compute $x_{n+1} = x_n - \alpha_n \nabla f_{j_n}(x_n)$.

3. Empirical Risk Minimization: The empirical risk is defined by

$$R(\phi) = \frac{1}{M}\sum_{m=1}^{M} \ell(\phi(x_m), y_m)).$$

Key assumption for convergence

We suppose that f is convex, $\hat{x}$ is a minimizer of f, and there exist $L, B \geq 0$ such that

$$\mathbb{E}_\xi \left[\|g(x,\xi)\|^2 \right] \leq L^2 \|x - \hat{x}\|^2 + B^2.$$

Key assumption for convergence

We suppose that f is convex, $\hat{x}$ is a minimizer of f, and there exist $L, B \geq 0$ such that

$$\mathbb{E}_\xi \left[\|g(x,\xi)\|^2 \right] \leq L^2 \|x - \hat{x}\|^2 + B^2.$$

Example

In the context of incremental gradient, this holds if each f_m is L_m-Lipschitz, attains its minimum at some $\hat{x}_m$, and we set

$$L^2 = \frac{2}{M} \sum_{m=1}^{M} L_m^2 \qquad \text{and} \qquad B^2 = \frac{2}{M} \sum_{m=1}^{M} L_m^2 \|\hat{x}_m - \hat{x}\|^2.$$

Convergence results I: $L = 0$

Set

$$\sigma_n = \sum_{k=0}^{n} \alpha_k, \quad \tau_n = \sum_{k=0}^{n} \alpha_k^2, \quad \text{and} \quad \overline{x}_n = \frac{1}{\sigma_n} \sum_{k=0}^{n} \alpha_k x_k.$$

Then, for each $n \geq 1$, we have

$$\mathbb{E}\big[f(\overline{x}_n) - \min(f)\big] \leq \frac{D_0^2 + \tau_n B}{2\sigma_n},$$

where $D_0 = \mathrm{dist}(x_0, S)$.

Convergence results II: L possibly nonzero

Assume f is $\mu-$strongly convex and $\alpha_n \equiv \alpha$. For each $n \geq 1$, we have

$$\mathbb{E}\left[\|x_n - x^*\|^2\right] \leq D_0^2 \left(1 - 2\alpha\mu + \alpha^2 L^2\right)^n + \frac{\alpha B^2}{2\mu - \alpha L^2}.$$

Convergence results II: L possibly nonzero

Assume f is $\mu-$strongly convex and $\alpha_n \equiv \alpha$. For each $n \geq 1$, we have

$$\mathbb{E}\left[\|x_n - x^*\|^2\right] \leq D_0^2 \left(1 - 2\alpha\mu + \alpha^2 L^2\right)^n + \frac{\alpha B^2}{2\mu - \alpha L^2}.$$

Questions

- Can we obtain a convergence rate for $\mathbb{E}\big(f(x_n) - \min(f)\big)$?

Convergence results II: L possibly nonzero

Assume f is $\mu-$strongly convex and $\alpha_n \equiv \alpha$. For each $n \geq 1$, we have

$$\mathbb{E}\left[\|x_n - x^*\|^2\right] \leq D_0^2 \left(1 - 2\alpha\mu + \alpha^2 L^2\right)^n + \frac{\alpha B^2}{2\mu - \alpha L^2}.$$

Questions

- Can we obtain a convergence rate for $\mathbb{E}\big(f(x_n) - \min(f)\big)$?
- If $B = 0$, we obtain linear convergence. What is the best possible rate? How does this compare with the deterministic gradient method?

Convergence results II: L possibly nonzero

Assume f is $\mu-$strongly convex and $\alpha_n \equiv \alpha$. For each $n \geq 1$, we have

$$\mathbb{E}\left[\|x_n - x^*\|^2\right] \leq D_0^2 \left(1 - 2\alpha\mu + \alpha^2 L^2\right)^n + \frac{\alpha B^2}{2\mu - \alpha L^2}.$$

Questions

- Can we obtain a convergence rate for $\mathbb{E}\big(f(x_n) - \min(f)\big)$?
- If $B = 0$, we obtain linear convergence. What is the best possible rate? How does this compare with the deterministic gradient method?
- If $B \neq 0$, can we obtain convergence by using vanishing step sizes?

Variants

- Batching for incremental gradient.

Variants

- Batching for incremental gradient.
- Nesterov's Acceleration (similar with heavy ball)

$$\begin{cases} y_n &= x_n + \beta_n(x_n - x_{n-1}) \\ x_{n+1} &= SG(y_n) \end{cases}$$

Variants

- Batching for incremental gradient.
- Nesterov's Acceleration (similar with heavy ball)

$$\begin{cases} y_n &= x_n + \beta_n(x_n - x_{n-1}) \\ x_{n+1} &= SG(y_n) \end{cases}$$

- Parameter selection
  - Step sizes (learning rates) prescribed *a priori*: Epochs
  - Adaptive: Adam

Variants

- Batching for incremental gradient.
- Nesterov's Acceleration (similar with heavy ball)

$$\begin{cases} y_n = x_n + \beta_n(x_n - x_{n-1}) \\ x_{n+1} = SG(y_n) \end{cases}$$

- Parameter selection
    - Step sizes (learning rates) prescribed *a priori*: Epochs
    - Adaptive: Adam
- Variance reduction
    - SVRG, SAG, SAGA

Adam: Adaptive Moment Estimation (2015)

- The direction is updated by

$$d_n = \left[\beta_1 d_{n-1} + (1-\beta_1) g_n\right](1-\beta_1^n)^{-1}.$$

Adam: Adaptive Moment Estimation (2015)

- The direction is updated by

$$d_n = \left[\beta_1 d_{n-1} + (1-\beta_1) g_n\right](1-\beta_1^n)^{-1}.$$

- The second order moment is estimated by

$$v_n^{(i)} = \left[\beta_2 v_{n-1}^{(i)} + (1-\beta_2)\left(g_n^{(i)}\right)^2\right](1-\beta_2^n)^{-1}$$

Adam: Adaptive Moment Estimation (2015)

- The direction is updated by

$$d_n = \big[\beta_1 d_{n-1} + (1-\beta_1) g_n\big](1-\beta_1^n)^{-1}.$$

- The second order moment is estimated by

$$v_n^{(i)} = \left[\beta_2 v_{n-1}^{(i)} + (1-\beta_2)\left(g_n^{(i)}\right)^2\right](1-\beta_2^n)^{-1}$$

- The step size (learning rate) is set at $\alpha_n^{(i)} = \dfrac{\alpha}{\sqrt{v_n^{(i)} + \varepsilon}}$.

Adam: Adaptive Moment Estimation (2015)

- The direction is updated by

$$d_n = \left[\beta_1 d_{n-1} + (1-\beta_1) g_n\right](1-\beta_1^n)^{-1}.$$

- The second order moment is estimated by

$$v_n^{(i)} = \left[\beta_2 v_{n-1}^{(i)} + (1-\beta_2)\left(g_n^{(i)}\right)^2\right](1-\beta_2^n)^{-1}$$

- The step size (learning rate) is set at $\alpha_n^{(i)} = \dfrac{\alpha}{\sqrt{v_n^{(i)} + \varepsilon}}$.

- Finally, the next iterate is computed by $x_{n+1} = x_n - \alpha_n d_n$.

Variance Reduction

- Motivation:
  - Let $\mathcal{X}, \mathcal{Y}$ be two random variables and set $\mathcal{Z} = \mathcal{X} - \big(\mathcal{Y} - \mathbb{E}(\mathcal{Y})\big)$.

Variance Reduction

- Motivation:
  - Let $\mathcal{X}, \mathcal{Y}$ be two random variables and set $\mathcal{Z} = \mathcal{X} - \big(\mathcal{Y} - \mathbb{E}(\mathcal{Y})\big)$.
  - Then, $\mathbb{E}(\mathcal{Z}) \sim \mathbb{E}(\mathcal{X})$ and $\mathbb{V}(\mathcal{Z}) = \mathbb{V}(\mathcal{X}) - 2\mathrm{Cov}(\mathcal{X}, \mathcal{Y}) + \mathbb{V}(\mathcal{Y})$.

Variance Reduction

- Motivation:
  - Let $\mathcal{X}, \mathcal{Y}$ be two random variables and set $\mathcal{Z} = \mathcal{X} - \big(\mathcal{Y} - \mathbb{E}(\mathcal{Y})\big)$.
  - Then, $\mathbb{E}(\mathcal{Z}) \sim \mathbb{E}(\mathcal{X})$ and $\mathbb{V}(\mathcal{Z}) = \mathbb{V}(\mathcal{X}) - 2\mathrm{Cov}(\mathcal{X}, \mathcal{Y}) + \mathbb{V}(\mathcal{Y})$.
  - If $\mathcal{X}, \mathcal{Y}$ are highly correlated, then $\mathbb{V}(\mathcal{Z})$ is small.

Variance Reduction

- Motivation:
  - Let $\mathcal{X}, \mathcal{Y}$ be two random variables and set $\mathcal{Z} = \mathcal{X} - \big(\mathcal{Y} - \mathbb{E}(\mathcal{Y})\big)$.
  - Then, $\mathbb{E}(\mathcal{Z}) \sim \mathbb{E}(\mathcal{X})$ and $\mathbb{V}(\mathcal{Z}) = \mathbb{V}(\mathcal{X}) - 2\mathrm{Cov}(\mathcal{X}, \mathcal{Y}) + \mathbb{V}(\mathcal{Y})$.
  - If $\mathcal{X}, \mathcal{Y}$ are highly correlated, then $\mathbb{V}(\mathcal{Z})$ is small.
- SAG (Stochastic Average Gradient):
  - Strongly convex case (2012).
  - Convex case (2014).
  - SAGA (2014): Unbiased, suitable for nonsmooth and non-strongly convex functions.

SAGA

We have $f(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x)$

If $j \in \{1, \ldots, M\}$ is picked uniformly at random, then

$$\mathbb{E}_j\big(\nabla f_j(x)\big) = \frac{1}{M}\sum_{m=1}^{M} \nabla f_m(x).$$

SAGA

We have $f(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x)$

If $j \in \{1, \ldots, M\}$ is picked uniformly at random, then

$$\mathbb{E}_j\big(\nabla f_j(x)\big) = \frac{1}{M}\sum_{m=1}^{M} \nabla f_m(x).$$

In the notation introduced above, we set

$$\begin{aligned}
\mathcal{X} &= \nabla f_{j_n}(x_{n+1}) \\
\mathcal{Y} &= \nabla f_{j_n}(x_n) \\
\mathcal{Z} &= \nabla f_{j_n}(x_{n+1}) - \left[\nabla f_{j_n}(x_n) - \frac{1}{M}\sum_{m=1}^{M} \nabla f_m(x_n)\right]
\end{aligned}$$

SAGA

We have $f(x) = \frac{1}{M}\sum_{m=1}^{M} f_m(x)$

Instead of

$$\mathcal{Z} = \nabla f_{j_n}(x_{n+1}) - \left[\nabla f_{j_n}(x_n) - \frac{1}{M}\sum_{m=1}^{M} \nabla f_m(x_n)\right],$$

which would be costly, we define

$$\mathcal{Z}' = \nabla f_{j_n}(x_{n+1}) - \left[\nabla f_{j_n}(x_n) - \frac{1}{M}\sum_{m=1}^{M} g_m^{(n)}\right],$$

where we have $g_m^{(n)}$ in storage for $m = 1, \ldots, M$.

SAGA in practice

Begin with $x_0 \in \mathbb{R}^N$, compute $g_m^{(0)} = \nabla f_m(x_0)$ for $m = 1, \ldots, M$, and construct a matrix $\mathcal{G}_0 = \begin{bmatrix} g_1^{(0)} & \cdots & g_M^{(0)} \end{bmatrix}$.

SAGA in practice

Begin with $x_0 \in \mathbb{R}^N$, compute $g_m^{(0)} = \nabla f_m(x_0)$ for $m = 1, \ldots, M$, and construct a matrix $\mathcal{G}_0 = \begin{bmatrix} g_1^{(0)} & \cdots & g_M^{(0)} \end{bmatrix}$.

After iteration n, we have a point x_n and a matrix $\mathcal{G}_n$.

SAGA in practice

Begin with $x_0 \in \mathbb{R}^N$, compute $g_m^{(0)} = \nabla f_m(x_0)$ for $m = 1, \ldots, M$, and construct a matrix $\mathcal{G}_0 = \begin{bmatrix} g_1^{(0)} & \cdots & g_M^{(0)} \end{bmatrix}$.

After iteration n, we have a point x_n and a matrix $\mathcal{G}_n$.

Pick $j_n \in \{1, \ldots, M\}$ uniformly at random, compute $g_{j_n}^{(n+1)} = \nabla f_{j_n}(x_n)$, and update $\mathcal{G}_n$ to $\mathcal{G}_{n+1}$ by replacing only the j_n-th column by $g_{j_n}^{(n+1)}$.

SAGA in practice

Begin with $x_0 \in \mathbb{R}^N$, compute $g_m^{(0)} = \nabla f_m(x_0)$ for $m = 1, \ldots, M$, and construct a matrix $\mathcal{G}_0 = \left[g_1^{(0)} \ \cdots \ g_M^{(0)} \right]$.

After iteration n, we have a point x_n and a matrix $\mathcal{G}_n$.

Pick $j_n \in \{1, \ldots, M\}$ uniformly at random, compute $g_{j_n}^{(n+1)} = \nabla f_{j_n}(x_n)$, and update $\mathcal{G}_n$ to $\mathcal{G}_{n+1}$ by replacing only the j_n-th column by $g_{j_n}^{(n+1)}$.

Finally, $x_{n+1} = x_n - \alpha_n \left[g_{j_n}^{(n+1)} - g_{j_n}^{(n)} + \frac{1}{M} \sum_{m=1}^{M} g_m^{(n)} \right]$, with $\alpha_n > 0$.