Gradient descent

Instructor: Jin Zhang

Department of Mathematics
Southern University of Science and Technology
Spring 2024

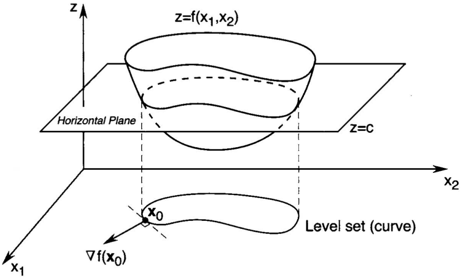
Main features of gradient methods

- The most popular methods (in continuous optimization)
- simple and intuitive
- work under very few assumptions

  (although they cannot directly handle nondifferentiable objectives and constraints, without applying smoothing techniques)
- work together with many other methods: duality, splitting, coordinate descent, alternating direction, stochastic, online, etc.
- suitable for large-scale problems, e.g., easy to parallelize for problems with many terms in the objective

Gradients

- We let $\nabla f(x_0)$ denote the gradient of f at point x_0.
- $\nabla f(x_0) \perp$ tangent of the levelset curve of f passing x_0, pointing outward (recall: level set $\mathcal{L}_f(c) := \{x : f(x) = c\}$)



- $\nabla f(x_0)$ is max-rate ascending direction of f at x_0 (for a small displacement), and $\|\nabla f(x_0)\|$ is the rate.
  Reason: pick any direction d with $\|d\| = 1$. The rate of change at x is

$$\left\langle \nabla f(x), d \right\rangle \leq \|\nabla f(x)\| \cdot \|d\| = \|\nabla f(x)\|.$$

  If we set $d = \nabla f(x) / \|\nabla f(x)\|$, then

$$\left\langle \nabla f(x), d \right\rangle = \|\nabla f(x)\|$$

- Therefore, $-\nabla f(x)$ is the max-rate descending direction of f and a good search direction.

A negative gradient step can decrease the objective

- Let $x^{(0)}$ be any initial point.
- First-order Taylor expansion for candidate point $x = x^{(0)} - \alpha \nabla f(x^{(0)})$:

$$f(x) - f(x^{(0)}) = -\alpha \|\nabla f(x^{(0)})\|^2 + o(\alpha)$$

- Hence, if $\nabla f(x^{(0)}) \neq 0$ (the first-order necessary condition is not met) and α is sufficiently small, we have

$$f(x) < f(x^{(0)}).$$

- Therefore, for sufficiently small α, x is an improvement over $x^{(0)}$.

Gradient descent algorithm

- Given initial $x^{(0)}$, the gradient descent algorithm uses the following update to generate $x^{(1)}, x^{(2)}, \cdots$, until a stopping condition is met: from the current point $x^{(k)}$, generate the next point $x^{(k+1)}$ by

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

- α_k is called the step size

Alternative interpretation:

- notice that

$$
\begin{aligned}
x^{(k+1)} &= \arg\min_x \frac{1}{2\alpha_k}\|x - x^{(k)} + \alpha_k \nabla f(x^{(k)})\|^2 \\
&= \arg\min_x f(x^{(k)}) + \langle \nabla f(x^{(k)}, x - x^{(k)}) \rangle + \frac{1}{2\alpha_k}\|x - x^{(k)}\|^2
\end{aligned}
$$

  (2nd "=" follows from that adding constants and multiplying a positive constant do not change the set of minimizers or "argmin")

- Hence, $x^{(k+1)}$ is obtained by minimizing the linearization of f at $x^{(k)}$ and a proximal term that keeps $x^{(k+1)}$ close to $x^{(k)}$.
- The reformulation is useful to develop the extensions of gradient descent:
  - projected gradient method
  - proximal-gradient method
  - accelerated gradient method
  - ......

When to stop the iteration

The first-order necessary condition $\|\nabla f(x^{(k+1)})\| = 0$ is not practical.Practical conditions:

- gradient condition $\|\nabla f(x^{(k+1)})\| < \epsilon$
- successive objective condition $|f(x^{(k+1)}) - f(x^{(k)})| < \epsilon$ or the relative one

$$\frac{|f(x^{(k+1)}) - f(x^{(k)})|}{|f(x^{(k)})|} < \epsilon$$

- successive point difference $\|x^{(k+1)} - x^{(k)}\| < \epsilon$ or the relative one

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} < \epsilon$$

- to avoid division by tiny numbers (unstable division), we can replace the denominators by $\max\{1, |f(x^{(k)})|\}$ and $\max\{1, \|x^{(k)}\|\}$, respectively

Small versus large step sizes α_k

Small step size:

- Pros: iterations are more likely converge, closely traces max-rate descends
- Cons: need more iterations and thus evaluations of ∇f

Large step size:

- Pros: better use of each $\nabla f(x^{(k)})$, may reduce the total iterations
- Cons: can cause overshooting and zig-zags, too large $\Rightarrow$ diverged iterations

Small versus large step sizes α_k

Small step size:

- Pros: iterations are more likely converge, closely traces max-rate descends
- Cons: need more iterations and thus evaluations of ∇f

Large step size:

- Pros: better use of each $\nabla f(x^{(k)})$, may reduce the total iterations
- Cons: can cause overshooting and zig-zags, too large $\Rightarrow$ diverged iterations

In practice, step sizes are often chosen

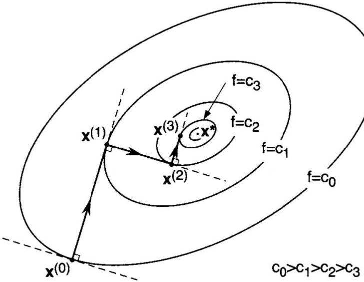
- as a fixed value if ∇f is Lipschitz (rate of change is bounded) with the constant known or an upper bound of it known
- by line search
- by a method called Barzilai-Borwein with nonmonotone line search

Steepest descent method
(gradient descent with exact line search)

Step size α_k is determined by exact minimization

$$\alpha_k = \operatorname*{argmin}_{\alpha \geq 0} f(x^{(k)} - \alpha \nabla f(x^{(k)})).$$

It is used mostly for quadratic programs (with α_k in a closed form) and some problems with inexpensive evaluation values but expensive gradient evaluation; otherwise it is not worth the effort to solve this subproblem exactly.



Proposition

If $\{x^{(k)}\}_{k=0}^{\infty}$ is a steepest descent sequence for a given function $f : \mathbb{R}^n \to \mathbb{R}$, then for each k the vector $x^{(k+1)} - x^{(k)}$ is orthogonal to the vector $x^{(k+2)} - x^{(k+1)}$.

Steepest descent for quadratic programming

Assume that the matrix Q is symmetric and positive definite ($x^T Qx > 0$ for any $x \neq 0$). Consider the quadratic program

$$f(x) = \frac{1}{2} x^T Qx - b^T x$$

with

$$\nabla f(x) = Qx - b.$$

Steepest descent iteration: start from any $x^{(0)}$, set

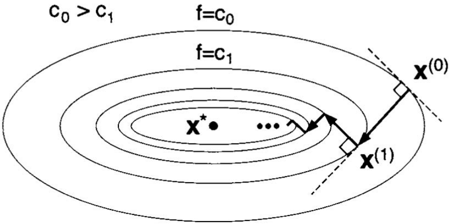
$$x^{(k+1)} = x^{(k)} - \alpha_k g^{(k)}, \ \ k = 0, 1, 2, \cdots$$

where $g^{(k)} := \nabla f\left(x^{(k)}\right)$ and

$$\alpha_k = \arg\min_{\alpha \geq 0} f\left(x^{(k)} - \alpha_k g^{(k)}\right) = \frac{g^{(k)\,T} g^{(k)}}{g^{(k)\,T} Q g^{(k)}}.$$

Examples

Example 1: $f(x) = x_1^2 + x_2^2$. Steepest descent arrives at $x^* = 0$ in 1 iteration.
Example 1: $f(x) = \frac{1}{5}x_1^2 + x_2^2$. Steepest descent makes progress in a narrow valley



Performance of steepest descent

- Per-iteration cost: dominated by two matrix-vector multiplications:
  - $g^{(k)} = Qx^{(k)} - b$
  - computing α_k involves $Qg^{(k)}$

  but they can be easily reduced to one matrix-vector multiplication.

- Convergence speed: determined by the initial point and the spectral condition of Q. To analyze them, we
  - define solution error: $e^{(k)} = x^{(k)} - x^*$ (not known, an analysis tool)
  - have property: $g^{(k)} = Qx^{(k)} - b = Qe^{(k)}$.

Good cases:

- $e^{(k)}$ is an eigenvector of Q with eigenvalue λ

$$
\begin{aligned}
e^{(k+1)} = e^{k} - \alpha_k g^{(k)} &= e^{(k)} - \frac{{g^{(k)}}^T g^{(k)}}{{g^{(k)}}^T Q g^{(k)}} (Q e^{(k)}) \\
&= e^{(k)} + \frac{{g^{(k)}}^T g^{(k)}}{\lambda {g^{(k)}}^T g^{(k)}} (-\lambda e^{(k)}) = 0.
\end{aligned}
$$

- Q has only one distinct eigenvalue (the level sets of Q are circles)

The general case: define $\|e\|_A := \sqrt{e^T A e}$ and $\kappa := \lambda_{\max}(Q)/\lambda_{\min}(Q)$, then we have

$$
\|e^{(k)}\|_A \le \left(\frac{\kappa - 1}{\kappa + 1} \right)^k \|e^{(0)}\|_A .
$$

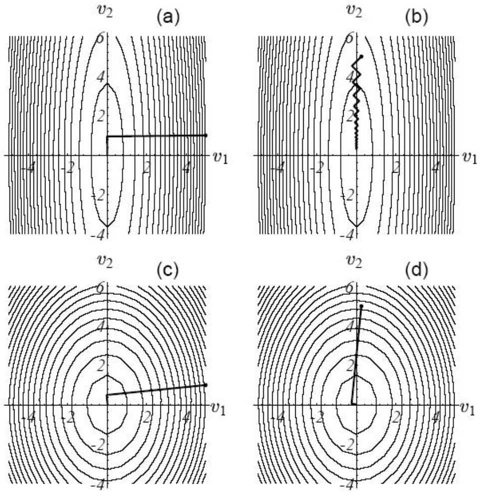


Figure: A example from An Introduction to CG method by Shewchuk

Gradient descent with fixed step size

- Iteration:
$$x^{(k+1)} = x^{(k)} - \alpha g^{(k)}$$
- We assume that x^* exists
- Check distance to solution:
$$\begin{aligned}\|x^{(k+1)} - x^*\|^2 &= \|x^{(k)} - x^* - \alpha g^{(k)}\|^2 \\ &= \|x^{(k)} - x^*\|^2 - 2\alpha\langle g^{(k)}, x^{(k)} - x^*\rangle + \alpha^2\|g^{(k)}\|^2.\end{aligned}$$
- Therefore, in order to have $\|x^{(k+1)} - x^*\| \leq \|x^{(k)} - x^*\|$, we must have
$$\frac{\alpha}{2}\|g^{(k)}\|^2 \leq \langle g^{(k)}, x^{(k)} - x^*\rangle.$$
Since $g^* := \nabla f(x^*) = 0$, the condition is equivalent to
$$\frac{\alpha}{2}\|g^{(k)} - g^*\|^2 \leq \langle g^{(k)} - g^*, x^{(k)} - x^*\rangle.$$

Special case: convex and Lipschitz differentiable f

- Definition: A function f is L-Lipschitz differentiable, $L \geq 0$, if $f \in \mathcal{C}^1$ and

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \forall x, y \in \mathbb{R}^n$$

(the maximum rate of change of ∇f is L)

- Baillon-Haddad theorem: if $f \in \mathcal{C}^1$ is a convex function, then it is L-Lipschitz differentiable if and only if

$$\|\nabla f(x) - \nabla f(y)\|^2 \leq L\langle \nabla f(x) - \nabla f(y), x - y\rangle.$$

(such ∇f is called $1/L$-cocoercive)

- **Theorem:** Let $f \in \mathcal{C}^1$ be a convex function and L-Lipschitz differentiable. If $0 < \alpha \le 2/L$, then

$$\frac{\alpha}{2}\|g^{(k)} - g^*\|^2 \le \langle g^{(k)} - g^*, x^{(k)} - x^* \rangle$$

  and thus $\|x^{(k+1)} - x^*\| \le \|x^{(k)} - x^*\|$ for $k = 0, 1 \cdots$ The iteration stays bounded.

- **Theorem:** Let $f \in \mathcal{C}^1$ be a convex function and L-Lipschitz differentiable. If $0 < \alpha < 2/L$, then
  - both $f(x^{(k)})$ and $\|\nabla f(x^{(k)})\|$ are monotonically decreasing,
  - $f(x^{(k)}) - f(x^{(*)}) = O(1/k)$,
  - $\|\nabla f(x^{(k)})\| = o(1/k)$.
    (one often writes $\|\nabla f(x^{(k)})\|^2 = o(1/k^2)$ since $\|\nabla f(x^{(k)})\|^2$ naturally appears in most analysis.)

Gradient descent with fixed step size for quadratic programming

Assume that Q is symmetric and positive definite ($x^T Qx > 0$ for any $x \neq 0$).

Consider the quadratic program

$$f(x) = \frac{1}{2}x^T Qx - b^T x.$$

Theorem

For the fixed-step-size gradient algorithm , $x^{(k)} \to x^*$ for any $x^{(0)}$ if and only if

$$0 < \alpha < \frac{2}{\lambda_{\max}(Q)}.$$

Summary

- Negative gradient $-\nabla f(x^{(k)})$ is the max-rate descending direction
- For some small α_k, $x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$ improves over $x^{(k)}$
- There are practical rules to determine when to stop the iteration
- Exact line search works for quadratic program with $Q > 0$. Zig-zag occurs if $x^{(0)} - x^*$ is away from an eigenvector and spectrum of Q is spread
- Fixed step gradient descent works for convex and Lipschitz-differentiable f
- To keep the discussion short and informative, we have omitted much other convergence analysis.