Math 273a: Optimization
Gradient descent

Instructor: Wotao Yin
Department of Mathematics, UCLA

Fall 2015

slides based on Chong-Zak, 4th Ed.

online discussions on piazza.com

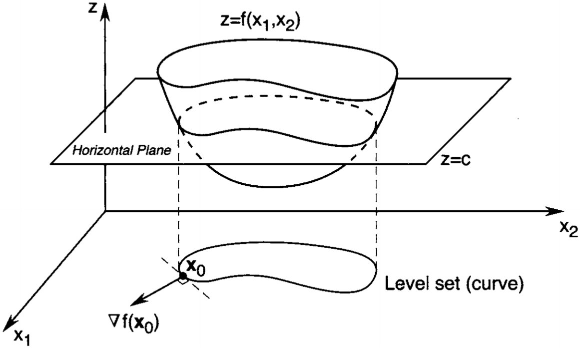
Main features of gradient methods

- 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 $\nabla f(\boldsymbol{x}_0)$ denote the gradient of f at point $\boldsymbol{x}_0$.
- $\nabla f(\boldsymbol{x}_0) \perp$ tangent of the levelset curve of f passing $\boldsymbol{x}_0$, pointing outward (recall: level set $\mathcal{L}_f(c) := \{\boldsymbol{x} : f(\boldsymbol{x}) = c\}$)



- $\nabla f(\boldsymbol{x}_0)$ is max-rate ascending direction of f at $\boldsymbol{x}_0$ (for a small displacement), and $\|\nabla f(\boldsymbol{x}_0)\|$ is the rate.

  Reason: pick any direction $\boldsymbol{d}$ with $\|\boldsymbol{d}\| = 1$. The rate of change at $\boldsymbol{x}$ is

  $$\langle \nabla f(\boldsymbol{x}), \boldsymbol{d} \rangle \leq \|\nabla f(\boldsymbol{x})\| \cdot \|\boldsymbol{d}\| = \|\nabla f(\boldsymbol{x})\|.$$

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

  $$\langle \nabla f(\boldsymbol{x}), \boldsymbol{d} \rangle = \|\nabla f(\boldsymbol{x})\|.$$

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

A negative gradient step can decrease the objective

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

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

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

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

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

Gradient descent algorithm

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

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

- α_k is called the step size

Alternative interpretation:

- notice that

$$\boldsymbol{x}^{(k+1)} = \arg\min_{\boldsymbol{x}} \frac{1}{2\alpha_k} \left\| \boldsymbol{x} - \left(\boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)}) \right) \right\|^2$$
$$= \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)}) + \langle \nabla f(\boldsymbol{x}^{(k)}), \boldsymbol{x} - \boldsymbol{x}^{(k)} \rangle + \frac{1}{2\alpha_k} \|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|^2$$

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

- Hence, $\boldsymbol{x}^{(k+1)}$ is obtained by minimizing the linearization of f at $\boldsymbol{x}^{(k)}$ and a proximal term that keeps x^{k+1} close to $\boldsymbol{x}^{(k)}$.

- 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 $\|\nabla f(\boldsymbol{x}^{(k+1)})\| = 0$ is not practical.

Practical conditions:

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

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

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

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

- to avoid division by tiny numbers (unstable division), we can replace the denominators by $\max\{1, |f(\boldsymbol{x}^{(k)})|\}$ and $\max\{1, \|\boldsymbol{x}^{(k)}\|\}$, 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 $\nabla f(\boldsymbol{x}^{(k)})$, 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 $\nabla f(\boldsymbol{x}^{(k)})$, may reduce the total iterations
- Cons: can cause overshooting and zig-zags, too large $\Rightarrow$ diverged iterations

In practice, step sizes are often chosen

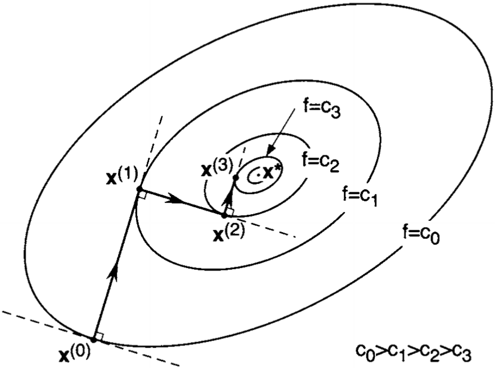
- as a fixed value if ∇f is Lipschitz (rate of change is bounded) with the constant known or an upper bound of it known
- by line search
- by a method called Barzilai-Borwein with nonmonotone line search

Steepest descent method
(gradient descent with exact line search)

Step size α_k is determined by exact minimization

$$\alpha_k = \arg\min_{\alpha \geq 0} \; f(\boldsymbol{x}^{(k)} - \alpha \nabla f(\boldsymbol{x}^{(k)})).$$

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

Steepest descent for quadratic programming

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

Consider the quadratic program

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

with

$$\nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{b}.$$

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

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

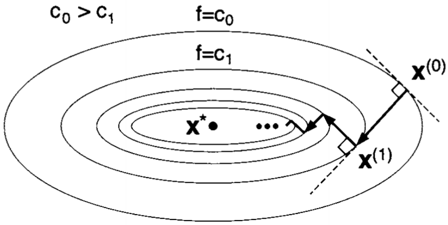
where $\boldsymbol{g}^{(k)} := \nabla f(\boldsymbol{x}^{(k)})$ and

$$\begin{aligned}\alpha_k &= \arg\min_{\alpha \geq 0} \ f(\boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)}) \\ &= \frac{\boldsymbol{g}^{(k)T}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)T}\boldsymbol{Q}\boldsymbol{g}^{(k)}}.\end{aligned}$$

Examples

Example 1: $f(\boldsymbol{x}) = x_1^2 + x_2^2$. Steepest descent arrives at $\boldsymbol{x}^* = \mathbf{0}$ in 1 iteration.

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



Performance of steepest descent

- **Per-iteration cost:** dominated by *two* matrix-vector multiplications:
  - $\boldsymbol{g}^{(k)} = \boldsymbol{Q}\boldsymbol{x}^{(k)} - \boldsymbol{b}$
  - computing α_k involves $\boldsymbol{Q}\boldsymbol{g}^{(k)}$

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

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

Good cases:

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

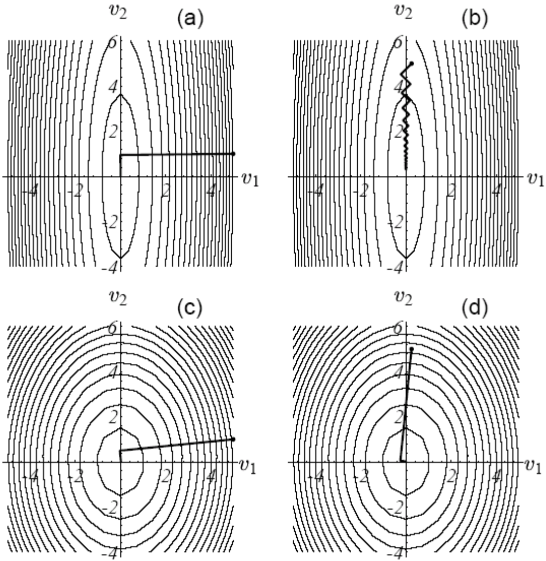
$$\boldsymbol{e}^{(k+1)} = \boldsymbol{e}_k - \alpha_k \boldsymbol{g}^{(k)} = \boldsymbol{e}^{(k)} - \frac{\boldsymbol{g}^{(k)T}\boldsymbol{g}^{(k)}}{\boldsymbol{g}^{(k)T}\boldsymbol{Q}\boldsymbol{g}^{(k)}}(\boldsymbol{Q}\boldsymbol{e}^{(k)})$$

$$= \boldsymbol{e}^{(k)} + \frac{\boldsymbol{g}^{(k)T}\boldsymbol{g}^{(k)}}{\lambda\,\boldsymbol{g}^{(k)T}\boldsymbol{g}^{(k)}}(-\lambda\boldsymbol{e}^{(k)}) = 0.$$

- $\boldsymbol{Q}$ has only one distinct eigenvalue (the level sets of $\boldsymbol{Q}$ are circles)

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

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



A example from *An Introduction to CG method* by Shewchuk

Gradient descent with fixed step size

- Iteration:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha \boldsymbol{g}^{(k)}$$

- We assume that $\boldsymbol{x}^*$ exists
- Check distance to solution:

$$\begin{aligned}\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\|^2 &= \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^* - \alpha \boldsymbol{g}^{(k)}\|^2 \\ &= \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|^2 - 2\alpha \langle \boldsymbol{g}^{(k)}, \boldsymbol{x}^{(k)} - \boldsymbol{x}^* \rangle + \alpha^2 \|\boldsymbol{g}^{(k)}\|^2.\end{aligned}$$

- Therefore, in order to have $\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^*\| \le \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\|$, we must have

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

  Since $\boldsymbol{g}^* := \nabla f(\boldsymbol{x}^*) = \mathbf{0}$, the condition is equivalent to

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

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

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

(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

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

(such ∇f is called $1/L$-cocoercive)

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

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

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

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

Gradient descent with fixed step size for quadratic programming

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

Consider the quadratic program

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

Theorem 8.3 *For the fixed-step-size gradient algorithm,* $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^*$ *for any* $\boldsymbol{x}^{(0)}$ *if and only if*

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

Summary

- Negative gradient $-\nabla f(\boldsymbol{x}^{(k)})$ is the max-rate descending direction
- For some small α_k, $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \nabla f(\boldsymbol{x}^{(k)})$ improves over $\boldsymbol{x}^{(k)}$
- There are practical rules to determine when to stop the iteration
- Exact line search works for quadratic program with $\boldsymbol{Q} > 0$. Zig-zag occurs if $\boldsymbol{x}^{(0)} - \boldsymbol{x}^*$ is away from an eigenvector and spectrum of $\boldsymbol{Q}$ 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.