Convex Optimization for Machine Learning and Computer Vision

Tutorial

12.12.2018

Contents

Overview

Problem settings

$$\text{minimize } J(u) \quad \text{over } u \in \mathbb{E}.$$

Assume:

1. $J : \mathbb{E} \to \mathbb{R}$ is continuously differentiable (C^1).
2. There exists a global minimizer u^*. (Typically, an optim algorithm seeks for a local minimizer s.t. $\nabla J(u^*) = 0$.)

Overview

Problem settings

$$\text{minimize } J(u) \quad \text{over } u \in \mathbb{E}.$$

Assume:

1. $J : \mathbb{E} \to \mathbb{R}$ is continuously differentiable (C^1).
2. There exists a global minimizer u^*. (Typically, an optim algorithm seeks for a local minimizer s.t. $\nabla J(u^*) = 0$.)

Gradient descent approach

1. Initialize $u^0 \in \mathbb{E}$ (often just randomly). Iterate ($k = 0, 1, 2, ...$) till convergence ($\|\nabla J(u^k)\| \leq \epsilon$):
2. Choose a **descent direction** $d^k \in \mathbb{E}$ s.t. $\left\langle \nabla J(u^k), d^k \right\rangle < 0$ and a **step size** $\tau^k > 0$, "Appropriately".
3. Update $u^{k+1} = u^k + \tau^k d^k$.

Choice of descent direction d^k and step size τ^k

How to choose descent direction?

Scaled gradient: $d^k = -(H^k)^{-1}\nabla J(u^k)$, H^k spd (why?).
Examples: Steepest descent: $H^k = I$; Newton (J is C^2): $H^k = \nabla^2 J(u^k)$ spd; Quasi-Newton: $H^k \approx \nabla^2 J(u^k)$ spd.

Choice of descent direction d^k and step size τ^k

How to choose descent direction?

Scaled gradient: $d^k = -(H^k)^{-1}\nabla J(u^k)$, H^k spd (why?).
Examples: Steepest descent: $H^k = I$; Newton (J is C^2): $H^k = \nabla^2 J(u^k)$ spd; Quasi-Newton: $H^k \approx \nabla^2 J(u^k)$ spd.

How to choose step size?

"Small enough": (Thm) ensures $J(u^k + \tau d^k) < J(u^k)$ (decrease of J) so long as $\left\langle \nabla J(u^k), d^k \right\rangle < 0$.
Exact line search: find the best τ^k along the direction, *often unrealistic!*
Inexact line search: find a good enough τ^k that ensures convergence.

- (A) Sufficient decrease condition (with $c_1 \in (0, 1)$)
- (C) Curvature condition (with $c_2 \in (c_1, 1)$)
- (A) $\rightsquigarrow$ **Armijo** line search; (A) & (C) $\rightsquigarrow$ **Wolfe-Powell** l.s.
- (Lemma) (A)+(C) is feasible; (Thm, Zoutendijk) "converge easily".

Backtracking (inexact) line search: some details

Why the name "backtracking"?

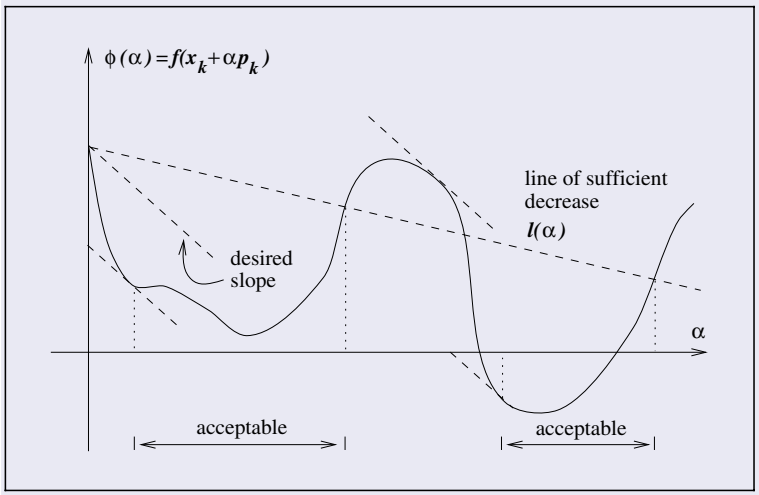
In practice, we start with a big estimate of τ^k and shrinks it until (A) + (C) are satisfied.

Backtracking (inexact) line search: some details

Why the name "backtracking"?

In practice, we start with a big estimate of τ^k and shrinks it until (A) + (C) are satisfied.

Thm. Zoutendijk: a closer look



(A) + (C) + J μ-Lipschitz diff. $\rightsquigarrow$

$$\sum_{k=0}^{\infty} \cos(\theta^k)^2 \|\nabla J(u^k)\|^2 < \infty.$$

with

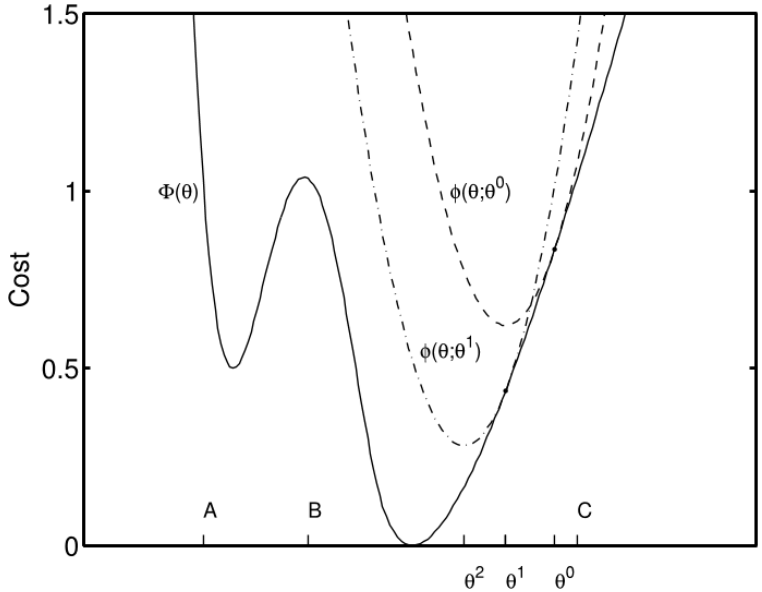
$$\cos(\theta^k) = \left\langle \nabla J(u^k), d^k \right\rangle / (\|\nabla J(u^k)\| \|d^k\|)$$

Remark: $-\cos(\theta^k) \geq c > 0 \Longrightarrow \lim_{k\to\infty} \|\nabla J(u^k)\| = 0$

i.e. **good enough direction ensures convergence!**

Majorize-minimize algorithm

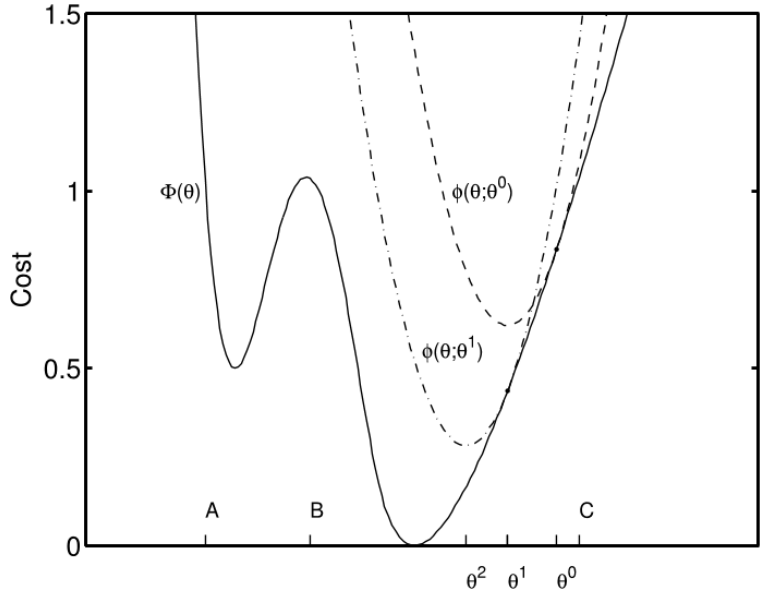
- **Main idea**: iteratively minimize an easy upper bound instead!
- **Majorant**: $\widehat{J}$ s.t. $\widehat{J}(\cdot\,; u)$ is a pointwise upper bound at $u \in \mathbb{E}$:
  1. $\widehat{J}(u; u) = J(u)$ (coincide at u);
  2. $\widehat{J}(\cdot\,; u) \geq J(\cdot)$ (upper bound at u).
- **Algorithm**: $u^{k+1} \in \arg\min_u \widehat{J}(u; u^k)$.



Monotonic Decrease: $J(u^{k+1}) \leq \widehat{J}(u^{k+1}; u^k) \leq \widehat{J}(u^k; u^k) = J(u^k)$.

Majorize-minimize algorithm

- **Main idea**: iteratively minimize an easy upper bound instead!
- **Majorant**: $\widehat{J}$ s.t. $\widehat{J}(\cdot\,; u)$ is a pointwise upper bound at $u \in \mathbb{E}$:
  1. $\widehat{J}(u; u) = J(u)$ (coincide at u);
  2. $\widehat{J}(\cdot\,; u) \geq J(\cdot)$ (upper bound at u).
- **Algorithm**: $u^{k+1} \in \arg\min_u \widehat{J}(u; u^k)$.



Monotonic Decrease: $J(u^{k+1}) \leq \widehat{J}(u^{k+1}; u^k) \leq \widehat{J}(u^k; u^k) = J(u^k)$.

Gradient descent as majorize-minimize algorithm

Let $J : \mathbb{E} \to \mathbb{R}$ is μ-Lipschitz diff., and $\tau \in (0, 1/\mu]$. Then we have:

- $\widehat{J}(u, u^k) = J(u^k) + \langle \nabla J(u^k), u - u^k \rangle + \frac{1}{2\tau}\|u - u^k\|^2$ is a majorant.
- $u^{k+1} = \arg\min_u \widehat{J}(u, u^k) \rightsquigarrow$ **convergent** GD: $\lim_{k\to\infty} \nabla J(u^k) = 0$.

Proximal gradient (Forward-Backward Splitting)

Problem settings

$$\min_u F(u) + G(u),$$

where G is convex differentiable but F is only convex, proper, lsc.
Remark: gradient descent not applicable: F might not be differentiable.

Proximal gradient (Forward-Backward Splitting)

Problem settings

$$\min_u F(u) + G(u),$$

where G is convex differentiable but F is only convex, proper, lsc.
Remark: gradient descent not applicable: F might not be differentiable.

Approach

Forward-backward splitting (FBS, or **proximal gradient**):

$$u^{k+1} = \text{prox}_{\tau F}(u^k - \tau \nabla G(u^k)) \quad (1)$$

$$= (I + \tau \partial F)^{-1} \circ (I - \tau \nabla G)(u^k). \quad (2)$$

How to ensure convergence?
Regularity condition on F, G and "appropriate" choice of τ, see later :).