# Generalized gradient descent

Barnabas Poczos & Ryan Tibshirani
Convex Optimization 10-725/36-725

## Recall subgradient method

We want to solve

$$\min_{x \in \mathbb{R}^n} \; f(x),$$

for $f$ convex, not necessarily differentiable

Subgradient method: choose initial $x^{(0)} \in \mathbb{R}^n$, and repeat

$$x^{(k)} = x^{(k-1)} - t_k \cdot g^{(k-1)}, \quad k = 1, 2, 3, \ldots,$$

where $g^{(k-1)}$ is any subgradient of $f$ at $x^{(k-1)}$

If $f$ is Lipschitz on a bounded set containing its minimizer, then subgradient method has convergence rate $O(1/\sqrt{k})$

Downside: can be very slow!

## Outline

Today:

- Generalized gradient descent
- Convergence analysis
- ISTA, matrix completion
- Special cases

## Decomposable functions

Suppose

$$f(x) = g(x) + h(x)$$

- $g$ is convex, differentiable
- $h$ is convex, not necessarily differentiable

If $f$ were differentiable, then gradient descent update would be:

$$x^+ = x - t \cdot \nabla f(x)$$

Recall motivation: minimize quadratic approximation to $f$ around $x$, replace $\nabla^2 f(x)$ by $\frac{1}{t} I$,

$$x^+ = \operatorname*{argmin}_z \underbrace{f(x) + \nabla f(x)^T (z - x) + \frac{1}{2t} \|z - x\|_2^2}_{\widehat{f}_t(z)}$$

In our case $f$ is not differentiable, but $f = g + h$, $g$ differentiable. Why don't we make quadratic approximation to $g$, leave $h$ alone?

I.e., update

$$\begin{aligned} x^+ &= \operatorname*{argmin}_z \; \widehat{g}_t(z) + h(z) \\ &= \operatorname*{argmin}_z \; g(x) + \nabla g(x)^T (z - x) + \frac{1}{2t}\|z - x\|_2^2 + h(z) \\ &= \operatorname*{argmin}_z \; \frac{1}{2t}\big\|z - \big(x - t\nabla g(x)\big)\big\|_2^2 + h(z) \end{aligned}$$

| | |
|---|---|
| $\frac{1}{2t}\big\|z - \big(x - t\nabla g(x)\big)\big\|_2^2$ | be close to gradient update for $g$ |
| $h(z)$ | also make $h$ small |

## Generalized gradient descent

Define

$$\operatorname{prox}_t(x) = \underset{z \in \mathbb{R}^n}{\operatorname{argmin}} \ \frac{1}{2t}\|x - z\|_2^2 + h(z)$$

Generalized gradient descent: choose initialize $x^{(0)}$, repeat

$$x^{(k)} = \operatorname{prox}_{t_k}\big(x^{(k-1)} - t_k \nabla g(x^{(k-1)})\big), \quad k = 1, 2, 3, \ldots$$

To make update step look familiar, can write it as

$$x^{(k)} = x^{(k-1)} - t_k \cdot G_{t_k}(x^{(k-1)})$$

where $G_t$ is the generalized gradient of $f$,

$$G_t(x) = \frac{x - \operatorname{prox}_t\big(x - t\nabla g(x)\big)}{t}$$

## What good did this do?

You have a right to be suspicious ... looks like we just swapped one minimization problem for another

Key point is that $\mathrm{prox}_t(\cdot)$ is can be computed analytically for a lot of important functions $h$. Note:

- $\mathrm{prox}_t$ doesn't depend on $g$ at all
- $g$ can be very complicated as long as we can compute its gradient

Convergence analysis: will be in terms of $\#$ of iterations of the algorithm

Each iteration evaluates $\mathrm{prox}_t(\cdot)$ once, and this can be cheap or expensive, depending on $h$

## ISTA

Consider lasso criterion

$$f(\beta) = \underbrace{\frac{1}{2}\|y - X\beta\|_2^2}_{g(\beta)} + \underbrace{\lambda\|\beta\|_1}_{h(\beta)}$$

Prox function is now

$$\begin{aligned} \mathrm{prox}_t(\beta) &= \underset{z\in\mathbb{R}^n}{\mathrm{argmin}}\ \frac{1}{2t}\|\beta - z\|_2^2 + \lambda\|z\|_1 \\ &= S_{\lambda t}(\beta) \end{aligned}$$

where $S_\lambda(\beta)$ is the soft-thresholding operator,

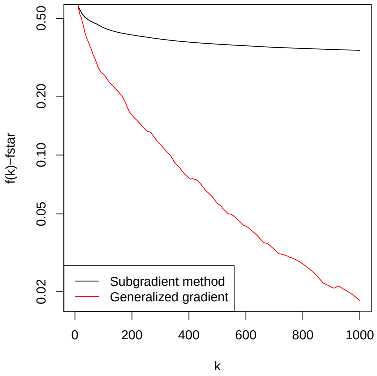
$$[S_\lambda(\beta)]_i = \begin{cases} \beta_i - \lambda & \text{if } \beta_i > \lambda \\ 0 & \text{if } -\lambda \le \beta_i \le \lambda \\ \beta_i + \lambda & \text{if } \beta_i < -\lambda \end{cases}$$

Recall $\nabla g(\beta) = -X^T(y - X\beta)$. Hence generalized gradient update step is:

$$\beta^+ = S_{\lambda t}\big(\beta + tX^T(y - X\beta)\big)$$

Resulting algorithm called ISTA (Iterative Soft-Thresholding Algorithm). Very simple algorithm to compute a lasso solution

Generalized gradient (ISTA) vs subgradient descent:



## Convergence analysis

We have $f(x) = g(x) + h(x)$, and assume

- $g$ is convex, differentiable, $\nabla g$ is Lipschitz continuous with constant $L > 0$
- $h$ is convex, $\mathrm{prox}_t(x) = \mathrm{argmin}_z\{\|x - z\|_2^2/(2t) + h(z)\}$ can be evaluated

**Theorem:** Generalized gradient descent with fixed step size $t \leq 1/L$ satisfies

$$f(x^{(k)}) - f(x^\star) \leq \frac{\|x^{(0)} - x^\star\|_2^2}{2tk}$$

I.e., generalized gradient descent has convergence rate $O(1/k)$

Same as gradient descent! But remember, this counts # of iterations, not # of operations

## Proof

Similar to proof for gradient descent, but with generalized gradient $G_t$ replacing gradient $\nabla f$. Main steps:

- $\nabla g$ Lipschitz with constant $L \Rightarrow$

$$f(y) \le g(x) + \nabla g(x)^T (y-x) + \frac{L}{2}\|y-x\|_2^2 + h(y) \quad \text{all } x,y$$

- Plugging in $y = x^+ = x - tG_t(x)$, and letting $t \le 1/L$,

$$f(x^+) \le g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2}\|G_t(x)\|_2^2 + h\big(x - tG_t(x)\big)$$

- By definition of prox,

$$x - tG_t(x) = \underset{z \in \mathbb{R}^n}{\operatorname{argmin}} \; \frac{1}{2t}\big\|z - \big(x - t\nabla g(x)\big)\big\|_2^2 + h(z)$$
$$\Rightarrow \quad \nabla g(x) - G_t(x) + v = 0, \quad v \in \partial h\big(x - tG_t(x)\big)$$

- Using $G_t(x) - \nabla g(x) \in \partial h\big(x - tG_t(x)\big)$, and convexity of $g$,

$$f(x^+) \le f(z) + G_t(x)^T(x-z) - \frac{t}{2}\|G_t(x)\|_2^2 \quad \text{all } z$$

- Letting $z = x$ verifies that generalized gradient steps indeed descend on the criterion. Letting $z = x^\star$,

$$\begin{aligned} f(x^+) &\le f(x^\star) + G_t(x)^T(x^\star - x) - \frac{t}{2}\|G_t(x)\|_2^2 \\ &= f(x^\star) + \frac{1}{2t}\big(\|x - x^\star\|_2^2 - \|x^+ - x^\star\|_2^2\big) \end{aligned}$$

Rest of proof follows as with gradient descent $\square$

## Backtracking line search

Similar to gradient descent, but operates on $g$ and not $f$

- Fix $0 < \beta < 1$
- Then at each iteration, start with $t = 1$, and while

$$g\big(x - tG_t(x)\big) > g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2}\|G_t(x)\|_2^2$$

  update $t = \beta t$

$$\boxed{\begin{array}{l}\textbf{Theorem: } \text{Generalized gradient descent with backtracking line search satisfies} \\ \qquad f(x^{(k)}) - f(x^\star) \le \dfrac{\|x^{(0)} - x^\star\|_2^2}{2t_{\mathsf{min}}k} \\ \text{where } t_{\mathsf{min}} = \min\{1, \beta/L\}\end{array}}$$

# Matrix completion

Given matrix $Y$, $m \times n$, only observe entries $Y_{ij}, (i,j) \in \Omega$

Want to fill in missing entries (e.g., NETFLIX ), so we solve:

$$\min_{B \in \mathbb{R}^{m \times n}} \frac{1}{2} \sum_{(i,j) \in \Omega} (Y_{ij} - B_{ij})^2 + \lambda \|B\|_*$$

Here $\|B\|_*$ is the nuclear norm of $B$,

$$\|B\|_* = \sum_{i=1}^{r} \sigma_i(B)$$

where $r = \mathrm{rank}(B)$ and $\sigma_1(B), \ldots \sigma_r(B)$ are its singular values

Define $P_\Omega$, projection operator onto observed set:

$$[P_\Omega(B)]_{ij} = \begin{cases} B_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$$

Criterion is

$$f(B) = \underbrace{\frac{1}{2}\|P_\Omega(Y) - P_\Omega(B)\|_F^2}_{g(B)} + \underbrace{\lambda\|B\|_*}_{h(B)}$$

Two ingredients for generalized gradient descent:

- Gradient: $\nabla g(B) = -(P_\Omega(Y) - P_\Omega(B))$
- Prox function:

$$\text{prox}_t(B) = \underset{Z \in \mathbb{R}^{m \times n}}{\text{argmin}} \; \frac{1}{2t}\|B - Z\|_F^2 + \lambda\|Z\|_*$$

Claim: $\mathrm{prox}_t(B) = S_{\lambda t}(B)$, where the matrix soft-thresholding operator $S_\lambda(B)$ is defined by

$$S_\lambda(B) = U\Sigma_\lambda V^T$$

Here $B = U\Sigma V^T$ is a singular value decomposition, and $\Sigma_\lambda$ is diagonal with

$$(\Sigma_\lambda)_{ii} = \max\{\Sigma_{ii} - \lambda, 0\}$$

Why? Note $\mathrm{prox}_t(B) = Z$, where $Z$ satisfies

$$0 \in Z - B + \lambda t \cdot \partial \|Z\|_*$$

Fact: if $Z = U\Sigma V^T$, then

$$\partial \|Z\|_* = \{UV^T + W : W \in \mathbb{R}^{m\times n}, \|W\| \leq 1, U^T W = 0, WV = 0\}$$

Now plug in $Z = S_{\lambda t}(B)$ and check that we can get $0$

Hence generalized gradient update step is:

$$B^+ = S_{\lambda t}\Big(B + t\big(P_\Omega(Y) - P_\Omega(B)\big)\Big)$$

Note that $\nabla g(B)$ is Lipschitz continuous with $L = 1$, so we can choose fixed step size $t = 1$. Update step is now:

$$B^+ = S_\lambda\big(P_\Omega(Y) + P_\Omega^\perp(B)\big)$$

where $P_\Omega^\perp$ projects onto unobserved set, $P_\Omega(B) + P_\Omega^\perp(B) = B$

This is the soft-impute algorithm[1], simple and effective method for matrix completion

[1]Mazumder et al. (2011), "Spectral regularization algorithms for learning large incomplete matrices"

## Why "generalized"?

Generalized gradient descent also called proximal gradient descent, or composite gradient descent

The descriptor "generalized" refers to the several special cases, when minimizing $f = g + h$:

- $h = 0 \rightarrow$ gradient descent
- $h = I_C \rightarrow$ projected gradient descent
- $g = 0 \rightarrow$ proximal minimization algorithm

Therefore these algorithms all have $O(1/k)$ convergence rate

## Projected gradient descent

Given closed, convex set $C \in \mathbb{R}^n$,

$$\min_{x \in C} \; g(x) \quad \Longleftrightarrow \quad \min_{x \in \mathbb{R}^n} \; g(x) + I_C(x)$$

where $I_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}$ is the indicator function of $C$

Hence

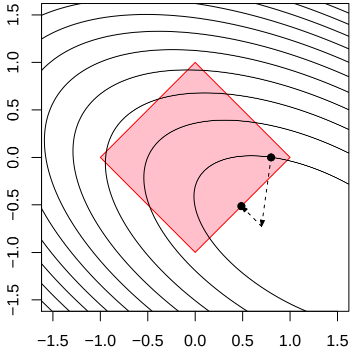
$$\begin{aligned} \operatorname{prox}_t(x) &= \operatorname*{argmin}_{z \in \mathbb{R}^n}, \frac{1}{2t} \|x - z\|_2^2 + I_C(z) \\ &= \operatorname*{argmin}_{z \in C} \; \|x - z\|_2^2 \end{aligned}$$

I.e., $\operatorname{prox}_t(x) = P_C(x)$, projection operator onto $C$

Therefore generalized gradient update step is:

$$x^+ = P_C\big(x - t\nabla g(x)\big)$$

i.e., perform usual gradient update and then project back onto $C$. Called projected gradient descent



## Proximal minimization algorithm

Consider for $h$ convex (not necessarily differentiable),

$$\min_{x \in \mathbb{R}^n} \; h(x)$$

Generalized gradient update step is just a prox update:

$$x^+ = \operatorname*{argmin}_{z \in \mathbb{R}^n} \; \frac{1}{2t}\|x - z\|_2^2 + h(z)$$

Called proximal minimization algorithm

Faster than subgradient method, but not implementable unless we know prox in closed form

## What happens if we can't evaluate prox?

Theory for generalized gradient, with $f = g + h$, assumes that prox function can be evaluated, i.e., assumes the minimization

$$\mathrm{prox}_t(x) = \underset{z \in \mathbb{R}^n}{\mathrm{argmin}} \; \frac{1}{2t}\|x - z\|_2^2 + h(z)$$

can be done exactly

Generally speaking, all bets are off if we just treat this as another minimization problem, and obtain an approximate solution. And practical convergence can be very slow if we use an approximation to the prox

But there are exceptions (both in theory and practice). E.g., if you can precisely control the errors in approximating the prox operator, then you can recover the original convergence rates[2]

[2]Schmidt et al. (2011), "Convergence rates of inexact proximal-gradient methods for convex optimization"

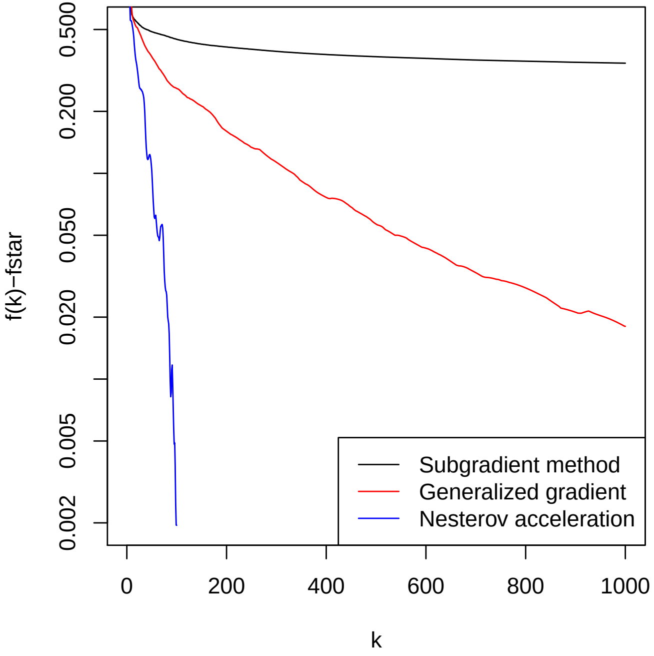
## Almost cutting edge

We're almost at the cutting edge for first order methods, but not quite ... still require too many iterations

Acceleration: use more than just $x^{(k-1)}$ to compute $x^{(k)}$ (e.g., use $x^{(k-2)}$), sometimes called momentum terms or memory terms

There are many different flavors of acceleration (at least three, mostly due to Nesterov)

Accelerated generalized gradient descent achieves optimal rate $O(1/k^2)$ among first order methods for minimizing $f = g + h$!


f(k)−fstar
0.500
0.200
0.050
0.020
0.005
0.002
0
200
400
600
800
1000
k
Subgradient method
Generalized gradient
Nesterov acceleration

## References

- E. Candes, Lecture Notes for Math 301, Stanford University, Winter 2010-2011
- L. Vandenberghe, Lecture Notes for EE 236C, UCLA, Spring 2011-2012