Chapter 6: Proximal Gradient Descent and Nesterov's Acceleration

Review

Consider the problem

$$\min_x f(x)$$

with f convex, and $\operatorname{dom}(f) = \mathbb{R}^n$.
Subgradient method:

- Choose an initial $x^{(0)} \in \mathbb{R}^n$
- Repeat:

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

  where $g^{(k-1)} \in \partial f\left(x^{(k-1)}\right)$.
  Use pre-set rules for the step sizes (e.g., fixed step sizes rule, diminshing step sizes rule)

> - If f is Lipschitz, then subgradient method has a convergence rate $\frac{1}{\sqrt{t}}$, and thus the complexity $O\left(1/\epsilon^2\right)$
> - Upside: very generic.
> - Downside: can be slow.

Composite functions

Suppose

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

where

- g is convex, differentiable, $\operatorname{dom}(g) = \mathbb{R}^n$
- h is convex, *not necessarily differentiable*.
  If f were differentiable, then gradient descent update would be:

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

> Minimize quadratic approximation to f around x, replace $\nabla^2 f(x)$ by $\frac{1}{t}I$
>
> $$x^+ = \underset{z}{\operatorname{argmin}} \underbrace{f(x) + \nabla f(x)^T(z - x) + \frac{1}{2t}\|z - x\|_2^2}_{\bar{f}_t(z)}$$

If f is not differentiable, but $f = g + h$, g differentiable. Consider the quadratic approximation to g but leave h alone.

> That is, update
>
> $$\begin{aligned} x^{+} &= \underset{z}{\operatorname{argmin}} \bar{g}_t(z) + h(z) \\ &= \underset{z}{\operatorname{argmin}}\ g(x) + \nabla g(x)^T(z-x) + \frac{1}{2t}\|z-x\|_2^2 + h(z) \\ &= \underset{z}{\operatorname{argmin}} \frac{1}{2t}\|z-(x-t\nabla g(x))\|_2^2 + h(z) \end{aligned}$$

Proximal gradient descent

Define ***proximal mapping***:

$$\operatorname{prox}_{h,t}(x) = \underset{z}{\operatorname{argmin}} \frac{1}{2t}\|x-z\|_2^2 + h(z)$$

Proximal gradient descent:

- Choose initialize $x^{(0)}$
- Repeat:

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

To make this update step look familiar, we can rewrite it as

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

where G_t can be seen as the *generalized gradient* of f,

$$G_t(x) = \frac{x - \operatorname{prox}_{h,t}(x - t\nabla g(x))}{t}$$

> $$\begin{aligned} x^{(k-1)} - t_k \cdot G_{t_k}\left(x^{(k-1)}\right) &= \operatorname{prox}_{h,t_k}\left(x^{(k-1)} - t_k\nabla g\left(x^{(k-1)}\right)\right) \\ G_{t_k}\left(x^{(k-1)}\right) &= \frac{x^{(k-1)} - \operatorname{prox}_{h,t_k}\left(x^{(k-1)} - t_k\nabla g\left(x^{(k-1)}\right)\right)}{t_k} \end{aligned}$$

- $\operatorname{prox}_{h,t}(\cdot)$ has a **closed-form** for many important functions h .
- Mapping prox $_{h,t}(\cdot)$ doesn't depend on g at all, only on h.
- Smooth part g can be complicated, we only need to compute its gradients.

Example: ISTA (iterative soft-thresholding algorithm) for lasso

Given $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n\times p}$, recall the lasso criterion:

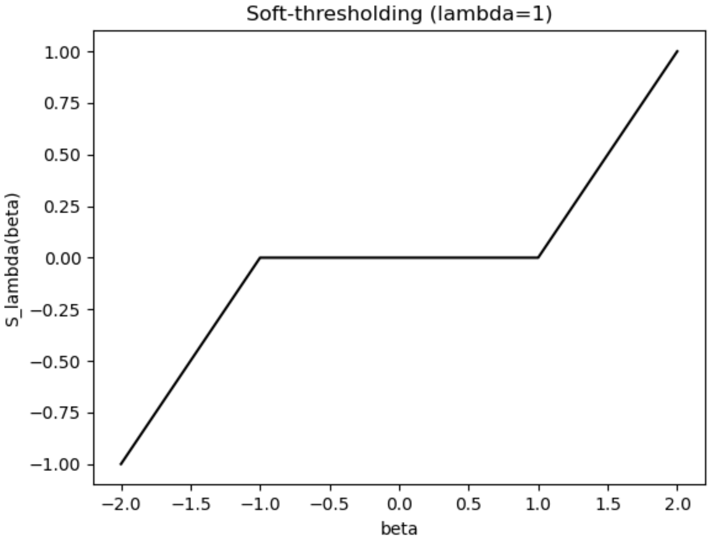
$$f(\beta) = \underbrace{\frac{1}{2}\|y - X\beta\|_2^2}_{g(\beta)} + \underbrace{\lambda\|\beta\|_1}_{h(\beta)}$$

Recall that the proximal mapping now is

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

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

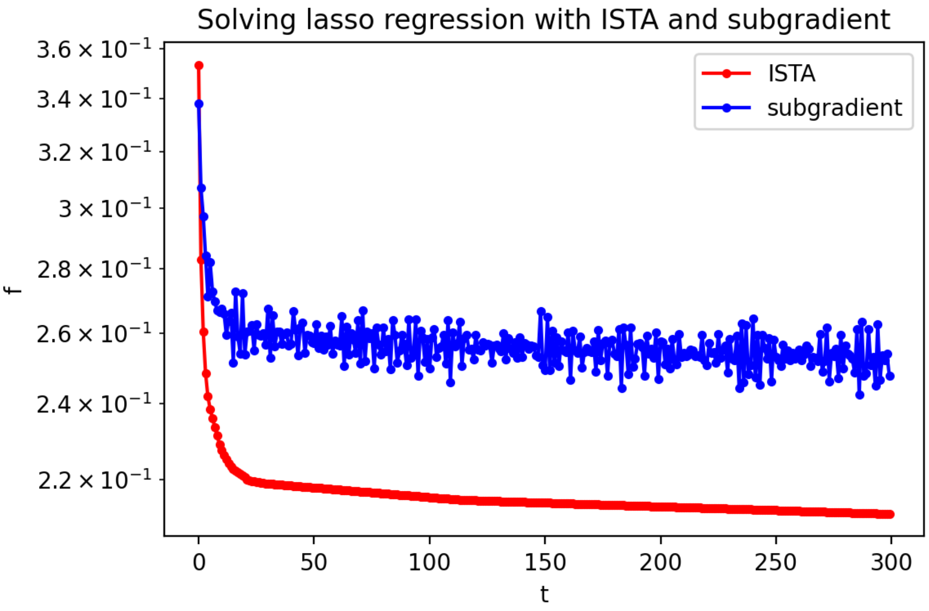
$$[S_\lambda(\beta)]_i = \begin{cases} \beta_i - \lambda & \text{if } \beta_i > \lambda \\ 0 & \text{if } -\lambda \le \beta_i \le \lambda, \\ \beta_i + \lambda & \text{if } \beta_i < -\lambda \end{cases} \quad i = 1, \dots, n$$



Recall $\nabla g(\beta) = -X^T(y - X\beta)$, hence proximal gradient update is:

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

Often called the **iterative soft-thresholding algorithm (ISTA)**[1].



Backtracking line search

Backtracking for proximal gradient descent works similar as in gradient descent, but operates on g and not f.

Choose parameter $0<\beta<1$. At each iteration, start at $t=t_{\text{init}}$ and while

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

shrink $t=\beta t$, for some $0<\beta<1$.

Convergence analysis

For criterion $f(x)=g(x)+h(x)$, where

- g is convex, differentiable, $\operatorname{dom}(g)=\mathbb{R}^n$, and ∇g is Lipschitz continuous with constant $L>0$.
- h is convex,.

Define

$$\operatorname{prox}_t(x)\triangleq\operatorname{argmin}_z\left\{\|x-z\|_2^2/(2t)+h(z)\right\}$$

which can be evaluated.

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

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

and same result holds for backtracking, with t replaced by β/L
Proximal gradient descent has convergence rate $O(1/k)$ or $O(1/\epsilon)$

Proof:

Due to the convexity and Lipschitz continuous of g, there holds

$$g(x^{(k+1)}) + h(x^{(k+1)}) \le g(x^{(k)}) + \langle x^{(k+1)} - x^{(k)}, \nabla g(x^{(k+1)})\rangle + \frac{L}{2}\|x^{(k+1)} - x^k\| + h(x^{(k+1)})$$

For any $z \in \mathbb{R}^n$, we have

$$f(z) - (g(x^{(k+1)}) + h(x^{(k+1)})) \ge f(z) - (g(x^k) + \langle x^{(k+1)} - x^{(k)}, \nabla g(x^k)\rangle + \frac{L}{2}\|x^{(k+1)} - x^k\| + h(x$$

Now, since g, h are convex we have

$$\begin{aligned} g(z) &\ge g(x^{(k)}) + \langle z - x^{(k)}, \nabla g(x^{(k)})\rangle \\ h(z) &\ge h(x^{(k+1)}) + \langle z - x^{(k+1)}, \gamma(x^{(k)})\rangle \end{aligned}$$

where $\gamma(x^{(k)}) \in \partial h(x^{(k)})$ such that

$$\gamma(x^k) = -\nabla f(x^{(k)}) - L(x^{(k+1)} - x^{(k)})$$

Summing the above two inequalities yields

$$f(x) \ge g(x^{(k)}) + \langle z - x^{(k)}, \nabla g(x^{(k)})\rangle + h(x^{(k+1)}) + \langle z - x^{(k+1)}, \gamma(x^{(k)})\rangle$$

Bring it to the first inequality gives

$$\begin{aligned} f(z) - f(x^{(k+1)}) &\ge -\frac{L}{2}\left\|x^{(k+1)} - x^{(k)}\right\|^2 + \left\langle z - x^{(k+1)}, \nabla g(x^{(k)}) + \gamma(x^{(k)})\right\rangle \\ &= -\frac{L}{2}\left\|x^{(k+1)} - x^{(k)}\right\|^2 + L\left\langle z - x^{(k+1)}, x^{(k)} - x^{(k+1)}\right\rangle \\ &= \frac{L}{2}\left\|x^{(k+1)} - x^{(k)}\right\|^2 + L\left\langle x^{(k)} - z, x^{(k+1)} - x^{(k)}\right\rangle \end{aligned}$$

This gives

$$\begin{aligned} \frac{2}{L}\left(f(x^\star) - f(x^{k+1})\right) &\ge \left\|x^{(k+1)} - x^{(k)}\right\|^2 + 2\left\langle x^{(k)} - x^\star, x^{(k+1)} - x^{(k)}\right\rangle \\ &= \left\|x^{(k+1)}\right\|^2 - \left\|x^{(k)}\right\|^2 - 2\left\langle x^\star, x^{(k+1)}\right\rangle + 2\left\langle x^\star, x^{(k)}\right\rangle \\ &= \left\|x^\star - x^{(k+1)}\right\|^2 - \left\|x^\star - x^{(k)}\right\|^2 \end{aligned}$$

Summing this inequality over $i = 0, \dots, k-1$ gives

$$\frac{2}{L}\left(kf(x^\star) - \sum_{i=0}^{k-1} f(x^{(i)})\right) \ge \left\|x^\star - x^k\right\|^2 - \left\|x^\star - x^0\right\|^2 \, (*)$$

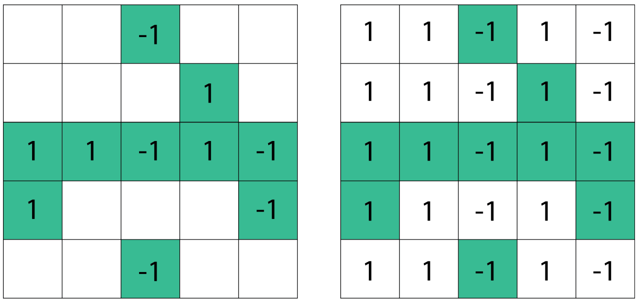
For

$$f(z) - f(x^{(k+1)}) \ge \frac{L}{2}\left\|x^{(k+1)} - x^{(k)}\right\|^2 + L\left\langle x^{(k)} - z, x^{(k+1)} - x^{(k)}\right\rangle$$

> Let $z = x^{(k)}$
>
> $$\frac{2}{L}\left(f(x^{(k)}) - f(x^{(k+1)})\right) \geq \left\|x^{(k)} - x^{(k+1)}\right\|^2$$
>
> Multiplying this inequality by i and summing over $i = 0, \ldots, k-1$, we obtain
>
> $$\frac{2}{L}\sum_{i=0}^{k-1}\left(if\left(x^{(i)}\right) - (i+1)f\left(x^{(i+1)}\right) + f\left(x^{(i+1)}\right)\right) \geq \sum_{i=0}^{k-1} i\left\|x^{(i)} - x^{(i+1)}\right\|^2$$
>
> $$\frac{2}{L}\left(-kf\left(x^{(k)}\right) + \sum_{i=0}^{k-1} f(x^{i+1})\right) \geq \sum_{i=0}^{k-1} i\left\|x^{(i)} - x^{(i+1)}\right\|^2$$
>
> Adding this inequality and $(*)$, we get
>
> $$\frac{2k}{L}\left(f\left(x^{\star}\right) - f\left(x^{(k)}\right)\right) \geq \left\|x^{\star} - x^{(k)}\right\|^2 + \sum_{i=0}^{k-1} i\left\|x^{(k)} - x^{(k+1)}\right\|^2 - \left\|x^{\star} - x^{(0)}\right\|^2$$
>
> and hence it follows that
>
> $$f\left(x^{(k)}\right) - f\left(x^{\star}\right) \leq \frac{L\left\|x^{\star} - x^{(0)}\right\|^2}{2k} \leq \frac{\left\|x^{\star} - x^{(0)}\right\|_2^2}{2tk}$$

Example: matrix completion

Given a matrix $Y \in \mathbb{R}^{m\times n}$, and only observe entries $Y_{ij}, (i,j) \in \Omega$. Suppose we want to fill in missing entries (e.g., for a recommender system).



		-1		
			1	
1	1	-1	1	-1
1				-1
		-1		

1	1	-1	1	-1
1	1	-1	1	-1
1	1	-1	1	-1
1	1	-1	1	-1
1	1	-1	1	-1

so we solve a matrix completion problem:

$$\min_{B} \frac{1}{2}\sum_{(i,j)\in\Omega}(Y_{ij} - B_{ij})^2 + \lambda\|B\|_{\text{tr}}$$

Here $\|B\|_{\text{tr}}$ is the trace (or nuclear) norm of B,

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

where $r = \operatorname{rank}(B)$ and $\sigma_1(X) \geq \cdots \geq \sigma_r(X) \geq 0$ are the singular values.

Define P_Ω, projection operator onto observed set:

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

Then the matrix completition problem is

$$\min_{B \in \mathbb{R}^{m \times n}} f(B) = \underbrace{\frac{1}{2} \|P_\Omega(Y) - P_\Omega(B)\|_F^2}_{g(B)} + \underbrace{\lambda \|B\|_{\text{tr}}}_{h(B)}$$

where the F-norm is $\|A\|_F^2 = \sum_{ij} A_{ij}^2$.

Two ingredients needed for proximal gradient descent:

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

$$\operatorname{prox}_t(B) = \underset{Z}{\operatorname{argmin}} \frac{1}{2t} \|B - Z\|_F^2 + \lambda \|Z\|_{\text{tr}}$$

Claim:

$$\operatorname{prox}_t(B) = S_{\lambda t}(B)$$

matrix soft-thresholding at the level λ.
Here $S_\lambda(B)$ is defined by

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

where $B = U \Sigma V^T$ is an SVD, and Σ_λ is diagonal with

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

> Proof: note that $\operatorname{prox}_t(B) = Z$, where Z satisfies
>
> $$0 \in Z - B + \lambda t \cdot \partial \|Z\|_{\text{tr}}$$
>
> If $Z = U \Sigma V^T$, then
>
> $$\partial \|Z\|_{\text{tr}} = \left\{UV^T + W : \|W\|_{\text{op}} \leq 1, U^T W = 0, WV = 0\right\}$$
>
> Now plug in $Z = S_{\lambda t}(B)$ and check that we can get 0.

Hence proximal gradient update step is:

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

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

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

where $P_\Omega^\perp$ projects onto unobserved set, $P_\Omega(B) + P_\Omega^\perp(B) = B$
This is the **soft-impute algorithm**[2].

Special cases

Proximal gradient descent also called **composite gradient descent** , or **generalized gradient descent**.
Here "generalized" refers to the several special cases:

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

Projected gradient descent

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

$$\min_{x \in C} g(x) \Longleftrightarrow \min_x g(x) + I_C(x)$$

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

Hence

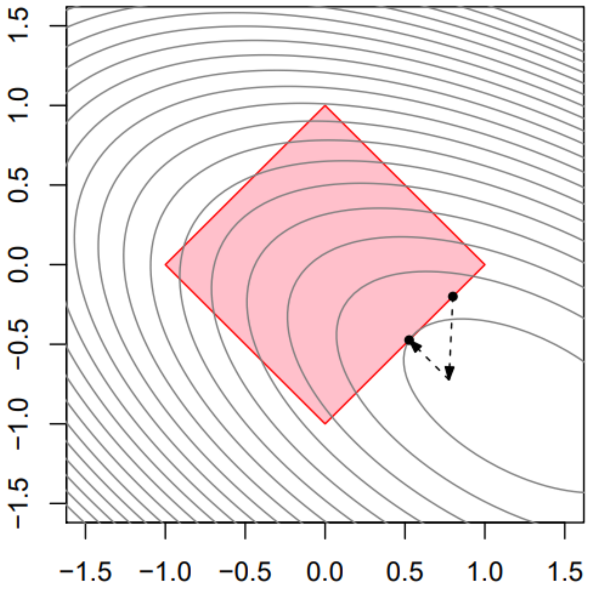
$$\begin{aligned} \operatorname{prox}_t(x) &= \underset{z}{\operatorname{argmin}} \frac{1}{2t} \|x - z\|_2^2 + I_C(z) \\ &= \underset{z \in C}{\operatorname{argmin}} \|x - z\|_2^2 \end{aligned}$$

That is, $\operatorname{prox}_t(x) = P_C(x)$, projection operator onto C

Therefore proximal gradient update step is:

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

That is to perform usual gradient update and then project back onto C. This is known as **projected gradient descent.**



Proximal minimization algorithm

What happens if we can't evaluate the prox?

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

$$\text{prox}_t(x) = \underset{z}{\text{argmin}} \frac{1}{2t}\|x - z\|_2^2 + h(z)$$

can be done exactly and quickly. In general, it is not clear what happens if we just minimize this approximately

But, if you can precisely control the errors in approximating the prox operator, then you may recover the original convergence rates.

In practice, if prox evaluation is done approximately, then it should be done to decently high accuracy.

Nesterov's Accelerated Gradient Method

Turns out we can accelerate proximal gradient descent in order to achieve the optimal $O(1/\sqrt{\epsilon})$ convergence rate. Four ideas (three acceleration methods) by Nesterov:

- 1983: original acceleration idea for smooth functions
- 1988: another acceleration idea for smooth functions
- 2005: smoothing techniques for nonsmooth functions, coupled with original acceleration idea
- 2007: acceleration idea for composite functions

We will follow Beck and Teboulle (2008)[1], an extension of Nesterov (1983) to composite functions.

Nesterov’s (momentum) acceleration

Consider the minimization of a convex f by the gradient descent method: $x^{(k)} = x^{(k-1)} - t_k \nabla f(x^{(k-1)})$.

In a seminal paper[3], ***Nesterov*** proposed an ***accelerated gradient method*** (Nesterov, 1983)

$$\begin{aligned} x^{(k)} &= v^{(k-1)} - t_k \nabla f\left(v^{(k-1)}\right) \\ v^{(k)} &= x^{(k)} + \tfrac{k-1}{k+2}\left(x^{(k)} - x^{(k-1)}\right) \end{aligned}$$

with initial $x^{(0)} = v^{(0)}$.

For any fixed step size $\eta \leq 1/L$, where L is the Lipschitz constant of ∇f, this scheme exhibits the convergence rate with constant step size t

$$f\left(x_k\right) - f^\star \leq O\left(\frac{\left\|x_0 - x^\star\right\|^2}{tk^2}\right)$$

Above, $x^\star$ is any minimizer of f and $f^\star = f\left(x^\star\right)$.

- It is well-known that this rate is optimal among all methods having only information about the gradient of f at consecutive iterates (Nesterov, 2004).
- This is in contrast to vanilla gradient descent methods, which have the same computational complexity but can only achieve a rate of $O(1/k)$.

Continuos model of Nesterove’s accelerated gradient method

- The continuous model of gradient descent is the **gradient flow** (an ordinary differential equation)

  $$X'(t) = -\nabla f(X)$$

  with $X(0) = x_0$

- In 2016, Su *et al* [4] shows the continuos ODE for the Nesterove’s accelerated gradient method takes the following form:

  $$\ddot{X} + \frac{3}{t}\dot{X} + \nabla f(X) = 0$$

  for $t > 0$, with initial conditions $X(0) = x_0, \dot{X}(0) = 0$; here, x_0 is the starting point in Nesterov’s scheme, $\dot{X} \equiv \mathrm{d}X/\mathrm{d}t$ denotes the time derivative or velocity .

(Nesterov’s) accelerated proximal gradient method

For the composite function

$$\min_x g(x) + h(x)$$

where g convex, differentiable, and h convex. Accelerated proximal gradient method: choose initial point $x^{(0)} = v^{(0)} \in \mathbb{R}^n$, repeat:

$$\begin{aligned} x^{(k)} &= \operatorname{prox}_{t_k}\left(v^{(k-1)} - t_k \nabla g(v^{(k-1)})\right) \\ v^{(k)} &= x^{(k)} + \frac{k-1}{k+2}\left(x^{(k)} - x^{(k-1)}\right) \end{aligned}$$

for $k = 1, 2, 3, \ldots$

> - First step $k = 1$ is just usual proximal gradient update
> - As $k \to \infty$, $\frac{k-1}{k+2} \to 1$ and $v^{(k)} \approx 2x^{(k)} - x^{(k-1)}$
> - After that, $v = x^{(k-1)} + \frac{k-2}{k+1}\left(x^{(k-1)} - x^{(k-2)}\right)$ carries some "**momentum**" from previous iterations
> - When $h = 0$ we get the Nesterov's accelerated gradient method.

Backtracking under with acceleration in different ways. Simple approach: fix $\beta < 1, t_0 = 1$. At iteration k, start with $t = t_{k-1}$, and while

$$g\left(x^+\right) > g(v) + \nabla g(v)^T\left(x^+ - v\right) + \frac{1}{2t}\left\|x^+ - v\right\|_2^2$$

shrink $t = \beta t$, and let $x^+ = \operatorname{prox}_t(v - t\nabla g(v))$. Else keep x^+

> This strategy forces us to take decreasing step sizes.

Convergence analysis

For criterion $f(x) = g(x) + h(x)$, we assume as before:

> - g is convex, differentiable, $\operatorname{dom}(g) = \mathbb{R}^n$, and ∇g is lipschitz continuous with constant $L > 0$.
> - h is convex, $\operatorname{prox}_t(x) = \operatorname{argmin}_z\left\{\|x - z\|_2^2/(2t) + h(z)\right\}$ can be evaluated.

Theorem: Accelerated proximal gradient method with fixed step size $t \leq 1/L$ satisfies

$$f\left(x^{(k)}\right) - f^\star \leq \frac{2\left\|x^{(0)} - x^\star\right\|_2^2}{t(k+1)^2}$$

> Proof: See Beck's paper[1].

Achieves optimal rate $O\left(1/k^2\right)$ or $O(1/\sqrt{\epsilon})$ for first-order methods.

Example: ISTA+Nesterov's acceleration

Back to lasso problem:

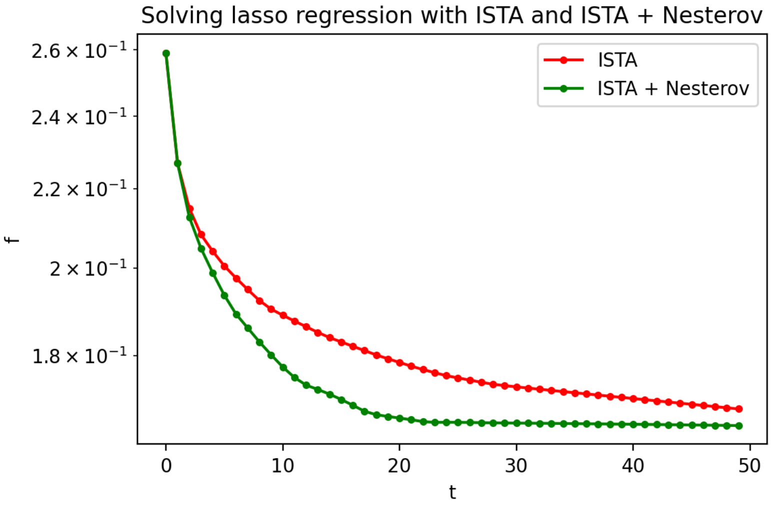
$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

Recall **ISTA** (Iterative Soft-thresholding Algorithm):

$$\beta^{(k)} = S_{\lambda t_k}\left(\beta^{(k-1)} + t_k X^T\left(y - X\beta^{(k-1)}\right)\right), \quad k = 1, 2, 3, \ldots$$

$S_\lambda(\cdot)$ being vector soft-thresholding. Applying acceleration gives us ISTA with **Nesterov's acceleration**:
For $k = 1, 2, 3, \ldots$

$$v = \beta^{(k-1)} + \frac{k-2}{k+1}\left(\beta^{(k-1)} - \beta^{(k-2)}\right)$$
$$\beta^{(k)} = S_{\lambda t_k}\left(v + t_k X^T(y - Xv)\right)$$



In practice the speedup of using acceleration is diminished in the presence of warm starts. For example, suppose want to solve lasso problem for tuning parameters values

$$\lambda_1 > \lambda_2 > \cdots > \lambda_r$$

> - When solving for λ_1, initialize $x^{(0)} = 0$, record solution $\hat{x}(\lambda_1)$.
> - When solving for λ_j, initialize $x^{(0)} = \hat{x}(\lambda_{j-1})$, the recorded solution for λ_{j-1}.
> - Over a fine enough grid of λ values, proximal gradient descent can often perform just as well without acceleration.

Sometimes backtracking and acceleration can be disadvantageous.
Recall matrix completion problem: the proximal gradient update is

$$B^+ = S_\lambda\left(B + t\left(P_\Omega(Y) - P^\perp(B)\right)\right)$$

where S_λ is the matrix soft-thresholding operator (requires SVD).
One backtracking loop evaluates prox, across various values of
t. For matrix completion, this means multiple SVDs.

> - Acceleration changes argument we pass to prox: $v - t\nabla g(v)$ instead of $x - t\nabla g(x)$.

For matrix completion (and $t = 1$).

$$B - \nabla g(B) = \underbrace{P_\Omega(Y)}_{\text{sparse}} + \underbrace{P_\Omega^\perp(B)}_{\text{low rank}} \Rightarrow \text{ fast SVD}$$

$$V - \nabla g(V) = \underbrace{P_\Omega(Y)}_{\text{sparse}} + \underbrace{P_\Omega^\perp(V)}_{\text{not necessarily low rank}} \Rightarrow \text{ slow SVD}$$

References and further reading

Nesterov's four ideas (three acceleration methods):

- Y. Nesterov (1983), "A method for solving a convex programming problem with convergence rate $O\left(1/k^2\right)$
- Y. Nesterov (1988), "On an approach to the construction of optimal methods of minimization of smooth convex functions "
- Y. Nesterov (2005), "Smooth minimization of non-smooth functions"
- Y. Nesterov (2007), "Gradient methods for minimizing composite objective function"
  Extensions and/or analyses:
- A. Beck and M. Teboulle (2008), "A fast iterative shrinkage-thresholding algorithm for linear inverse problems"
- S. Becker and J. Bobin and E. Candes (2009), "NESTA: a fast and accurate first-order method for sparse recovery"
  P. Tseng (2008), "On accelerated proximal gradient methods for convex-concave optimization"

Helpful lecture notes/books:
E. Candes, Lecture notes for Math 301, Stanford University, Winter 2010-2011

- Y. Nesterov (1998), "Introductory lectures on convex optimization: a basic course", Chapter 2
- Vandenberghe, Lecture notes for EE 236 C, UCLA, Spring 2011-2012

---

1. https://epubs.siam.org/doi/abs/10.1137/080716542?mobileUi=0 ↩ ↩ ↩

2. https://www.jmlr.org/papers/v11/mazumder10a.html ↩

3. https://www.semanticscholar.org/paper/A-method-for-solving-the-convex-programming-problem-Nesterov/8d3a318b62d2e970122da35b2a2e70a5d12cc16f ↩

4. https://jmlr.org/papers/v17/15-084.html ↩