Nonsmooth optimization

Most of the theory and algorithms that we have explored for convex optimization have assumed that the functions involved are differentiable – that is, smooth.

This is not always the case in interesting applications. In fact, nonsmooth functions can arise quite naturally in applications. We already have looked at optimization programs involving the hinge loss $\max(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x} + b, 0)$, the ℓ_1 norm, the ℓ_∞ norm, and the nuclear norm — none of these is differentiable. As another example, suppose $f_1, \ldots, f_Q$ are all perfectly smooth convex functions. Then the pointwise maximum

$$f(\boldsymbol{x}) = \max_{1 \le q \le Q} \; f_q(\boldsymbol{x})$$

is in general not smooth.

Fortunately, the theory for nonsmooth optimization is not too different than for smooth optimization. We really just need one new concept: that of a subgradient.

Subgradients

If you look back through the notes so far, you will see that the vast majority of the time we use the gradient of a convex function, it is in the context of the inequality

$$f(\boldsymbol{y}) \;\geq\; f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \nabla f(\boldsymbol{x}) \rangle, \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} f.$$

This is a very special property of convex functions, and it led to all kinds of beautiful results.

When a convex f is not differentiable at a point $\boldsymbol{x}$, we can more or less reproduce the entire theory using subgradients. A *subgradient* of f at $\boldsymbol{x}$ is a vector $\boldsymbol{g}$ such that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{g} \rangle, \quad \text{for all } \boldsymbol{y} \in \operatorname{dom} f.$$

Unlike gradients for smooth functions, there can be more than one subgradient of a nonsmooth function at a point. We call the collection of subgradients the *subdifferential* at $\boldsymbol{x}$:

$$\partial f(\boldsymbol{x}) = \{\boldsymbol{g} \;:\; f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{g} \rangle, \quad \text{for all } \boldsymbol{y} \in \operatorname{dom} f\}.$$

Facts:

1. If f is convex and differentiable at $\boldsymbol{x}$, then the subdifferential contains exactly one vector: the gradient,
$$\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}.$$
2. If f is convex on $\operatorname{dom} f$, then the subdifferential is non-empty and bounded at all $\boldsymbol{x}$ in the interior of $\operatorname{dom} f$.

For non-convex f, these two points do not hold in general. The gradient at a point is not necessarily a subgradient

and there can also be points where neither the gradient nor subgradient exist, e.g. $f(x) = -\sqrt{|x|}$ for $x \in \mathbb{R}$

Example: the ℓ_1 norm

Consider the function

$$f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1.$$

The ℓ_1 norm is not differentiable at any $\boldsymbol{x}$ that has at least one coordinate equal to zero. We will see that optimization problems involving the ℓ_1 norm very often have solutions that are sparse, meaning that they have many zeros. This is a big problem – the nonsmoothness is kicking in at exactly the points we are interested in.

What does the subdifferential $\partial\|\boldsymbol{x}\|_1$ look like in such a case? To see, recall that by definition, if a vector $\boldsymbol{u} \in \partial\|\boldsymbol{x}\|_1$, at the point $\boldsymbol{x}$, then we must have

$$\|\boldsymbol{y}\|_1 \geq \|\boldsymbol{x}\|_1 + \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{u} \rangle \tag{1}$$

for all $\boldsymbol{y} \in \mathbb{R}^N$. To understand what this means in terms of $\boldsymbol{x}$, it is useful to introduce the notation $\Gamma(\boldsymbol{x})$ to denote the set of indexes where $\boldsymbol{x}$ is non-zero:

$$\Gamma(\boldsymbol{x}) = \{n \,:\, x_n \neq 0\}.$$

Using this, we can re-write the right-hand side of (1) as

$$\begin{aligned}\|\boldsymbol{x}\|_1 + \langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{u} \rangle &= \sum_{n=1}^{N} |x_n| + \sum_{n=1}^{N} u_n(y_n - x_n) \\ &= \sum_{n\in\Gamma} |x_n| - u_n x_n + \sum_{n=1}^{N} u_n y_n.\end{aligned}$$

Note that if

$$u_n = \operatorname{sign}(x_n) = \begin{cases} 1 & \text{if } x_n \geq 0, \\ -1 & \text{if } x_n < 0, \end{cases}$$

then $u_n x_n = |x_n|$. Thus, if $u_n = \operatorname{sign}(x_n)$ for all $n \in \Gamma$, we have

$$\sum_{n\in\Gamma} |x_n| - u_n x_n = \sum_{n\in\Gamma} |x_n| - |x_n| = 0.$$

Thus, if we set $u_n = \operatorname{sign}(x_n)$ for all $n \in \Gamma$, then (1) reduces to

$$\|\boldsymbol{y}\|_1 \geq \langle \boldsymbol{y}, \boldsymbol{u} \rangle.$$

As long as $|u_n| \leq 1$ for all n, then this will hold. Hence, if a vector $\boldsymbol{u}$ satisfies

$$\begin{aligned} u_n &= \operatorname{sign}(x_n) && \text{if } n \in \Gamma, \\ |u_n| &\leq 1 && \text{if } n \notin \Gamma, \end{aligned}$$

then $\boldsymbol{u} \in \partial\|\boldsymbol{x}\|_1$. It is not hard to show that for any $\boldsymbol{u}$ that violates these conditions, we can construct a $\boldsymbol{y}$ such that (1) is violated, and thus this is a complete description of all vectors in $\boldsymbol{u} \in \partial\|\boldsymbol{x}\|_1$.

Optimality conditions for unconstrained optimization

(New and Improved!!)

With the right definition in place, it is very easy to re-derive the central mathematical results in this course for general[1] convex functions.

> Let $f(\boldsymbol{x})$ be a general convex function. Then $\boldsymbol{x}^\star$ is a solution to the unconstrained problem
>
> $$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \;\; f(\boldsymbol{x})$$
>
> if and only if
>
> $$\mathbf{0} \in \partial f(\boldsymbol{x}^\star).$$

The proof of this statement is so easy you could do it in your sleep. Suppose $\mathbf{0} \in \partial f(\boldsymbol{x}^\star)$. Then

$$\begin{aligned} f(\boldsymbol{y}) &\geq f(\boldsymbol{x}^\star) + \langle \boldsymbol{y} - \boldsymbol{x}^\star, \mathbf{0}\rangle \\ &= f(\boldsymbol{x}^\star) \end{aligned}$$

for all $\boldsymbol{y} \in \operatorname{dom} f$. Thus $\boldsymbol{x}^\star$ is optimal. Likewise, if $f(\boldsymbol{y}) \geq f(\boldsymbol{x}^\star)$ for all $\boldsymbol{y} \in \operatorname{dom} f$, then of course it must also be true that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}^\star) + \langle \boldsymbol{y} - \boldsymbol{x}, \mathbf{0}\rangle$ for all $\boldsymbol{y}$, and so $\mathbf{0} \in \partial f(\boldsymbol{x}^\star)$.

[1] Meaning not necessarily differentiable.

Example: the LASSO

Consider the ℓ_1 regularized least-squares problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}}\ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\tau\|\boldsymbol{x}\|_1.$$

We can quickly translate the general result $\mathbf{0}\in\partial f(\boldsymbol{x}^\star)$ into a useful set of optimality conditions. We need to compute the subdifferential of $f(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2+\tau\|\boldsymbol{x}\|_1$. The first term is smooth, so the subdifferential just contains the gradient:

$$\partial f(\boldsymbol{x})=\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y})+\tau\partial\|\boldsymbol{x}\|_1.$$

As shown above $\partial\|\boldsymbol{x}\|_1$ is the set of all vectors $\boldsymbol{u}$ such that

$$\begin{aligned} u_n &= \operatorname{sign}(x_n) && \text{if } x_n\neq 0,\\ |u_n| &\leq 1 && \text{if } x_n=0. \end{aligned}$$

Thus the optimality condition

$$\mathbf{0}\in\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^\star-\boldsymbol{y})+\tau\partial\|\boldsymbol{x}^\star\|_1,$$

means that $\boldsymbol{x}^\star$ is optimal if and only if

$$\begin{aligned} \boldsymbol{a}_i^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}^\star) &= \tau\operatorname{sign}x^\star[i] && \text{if } x_n^\star\neq 0,\\ |\boldsymbol{a}_i^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}^\star)| &\leq \tau && \text{if } x_n^\star=0. \end{aligned}$$

where $\boldsymbol{a}_i^{\mathrm{T}}$ is the i^{th} column of $\boldsymbol{A}$.

Note that this doesn't quite give us a closed form expression for $\boldsymbol{x}^\star$ (except when $\boldsymbol{A}$ is an orthonormal matrix), but it is useful both algorithmically (for checking if a candidate $\boldsymbol{x}$ is a solution) and theoretically (for understanding and analyzing the properties of the solution to this optimization problem.)

The subgradient method

The subgradient method is the non-smooth version of gradient descent. The basic algorithm is straightforward, consisting of the iteration

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{d}_k, \tag{2}$$

where $\boldsymbol{d}_k$ is *any* subgradient at $\boldsymbol{x}_k$, i.e., $\boldsymbol{d}_k \in \partial f(\boldsymbol{x}_k)$. Of course, there could be many choices for $\boldsymbol{d}_k$ at every step, and the progress you make at that iteration could very dramatically with this choice. Making this determination, though, is often very difficult, and whether or not it can even be done it very problem dependent. Thus the analytical results for the subgradient method just assume we have any subgradient at a particular step.

With the right choice of step sizes $\{\alpha_k\}$, some simple analysis (which we will get to in a minute) shows that the subgradient method converges. The convergence rate, though, is very slow. This is also evidenced in most practical applications of this method: it can take many iterations on even a medium-sized problem to arrive at a solution that is even close to optimal.

Here is what we know about this algorithm for solving the general unconstrained program

$$\underset{\boldsymbol{x} \in \mathbb{R}^N}{\text{minimize}}\, f(\boldsymbol{x}). \tag{3}$$

We will just state the results here; for detailed derivations, see [**?**, Chapter 3]. Along with f being convex, we will assume that it has at least one minimizer. The results also assume that f is Lipschitz:

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \ \leq \ L\|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

Note that here we are assuming that f is Lipschitz, not that f has Lipshitz gradients (since the gradient does not even necessarily exist).

A direct consequence of f being Lipshitz is that the norms of the subgradients are bounded:

$$\|\boldsymbol{d}\|_2 \leq L, \quad \text{for all } \boldsymbol{d} \in \partial f(\boldsymbol{x}), \quad \text{for all } \boldsymbol{x} \in \mathbb{R}^N. \tag{4}$$

The results below used pre-determined step sizes. Thus the iteration (2) does not necessarily decrease the functional $f(\boldsymbol{x})$ at every step. We will keep track of the best value we have up to the current iteration with

$$f_k^{\text{best}} = \min\left\{f(\boldsymbol{x}_i), \quad 0 \leq i < k\right\}.$$

We will let $\boldsymbol{x}^\star$ be any solution to (3) and set $f^\star = f(\boldsymbol{x}^\star)$.

Our analytical results stem from a careful look at what happens during a single iteration. Note that

$$\begin{aligned}\|\boldsymbol{x}_{i+1} - \boldsymbol{x}^\star\|_2^2 &= \|\boldsymbol{x}_i - \alpha_i\boldsymbol{d}_i - \boldsymbol{x}^\star\|_2^2 \\ &= \|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 - 2\alpha_i\langle\boldsymbol{x}_i - \boldsymbol{x}^\star, \boldsymbol{d}_i\rangle + \alpha_i^2\|\boldsymbol{d}_i\|_2^2 \\ &\leq \|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 - 2\alpha_i(f(\boldsymbol{x}_i) - f^\star) + \alpha_i^2\|\boldsymbol{d}_i\|_2^2,\end{aligned}$$

where the inequality follows from the definition of a subgradient:

$$f^\star \geq f(\boldsymbol{x}_i) + \langle\boldsymbol{x}^\star - \boldsymbol{x}_i, \boldsymbol{d}_i\rangle.$$

Rearranging the bound above we have

$$2\alpha_i\left(f(\boldsymbol{x}_i) - f^\star\right) \leq \|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_{i+1} - \boldsymbol{x}^\star\|_2^2 + \alpha_i^2\|\boldsymbol{d}_i\|_2^2,$$

and so of course

$$2\alpha_i\left(f_{\text{best}}^{(i)} - f^\star\right) \leq \|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 + \alpha_i^2\|\boldsymbol{d}_i\|_2^2.$$

Since f_i^{best} is monotonically decreasing, at iteration k we have

$$2\alpha_i \left(f_k^{\text{best}} - f^\star\right) \le \|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_{i+1} - \boldsymbol{x}^\star\|_2^2 + \alpha_i^2 \|\boldsymbol{d}_i\|_2^2,$$

for all $i \le k$. To understand what has happened after k iterations, we sum both sides of the expression above from $i = 0$ to $i = k - 1$. Notice that the two error terms on the right hand side give us the telescoping sum:

$$\begin{aligned}\sum_{i=0}^{k-1} \left(\|\boldsymbol{x}_i - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_{i+1} - \boldsymbol{x}^\star\|_2^2\right) &= \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2 - \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|_2^2 \\ &\le \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2\end{aligned}$$

and so

$$f_k^{\text{best}} - f^\star \le \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2 + \sum_{i=0}^{k-1} \alpha_i^2 \|\boldsymbol{d}_i\|_2^2}{2\sum_{i=0}^{k-1} \alpha_i} \tag{5}$$

We can now specialize this result to general step-size strategies.

Fixed step size. Suppose that $\alpha_k = \alpha > 0$ for all k. Then (5) becomes

$$f_k^{\text{best}} - f^\star \le \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{2k\alpha} + \frac{L^2\alpha}{2},$$

where we have also used the Lipschitz property (4). Note that in this case, no matter how small we choose t, **the subgradient algorithm is not guaranteed to converge**. This is, in fact, standard in practice as well. The problem is that, unlike gradients for smooth functions, the subgradients do not have to vanish as we approach the solution. Even at the solution, there can be subgradients that are large.

Fixed step length. A similar result holds if we always move the same amount, taking

$$\alpha_k = s/\|\boldsymbol{d}_k\|_2.$$

This means that

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2 = s.$$

Of course, with this strategy it is self-evident that it will never converge, since we move some fixed amount at every step. We can bound the suboptimality at step k as

$$f_k^{\text{best}} - f^\star \leq \frac{G\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{2ks} + \frac{Ls}{2},$$

which is not necessarily worse than the fixed step size result. In fact, notice that even though you are moving some fixed amount, you will never move to0 far from an optimal point.

Decreasing step size. The results above suggest that we might want to decrease the step size as k increases, so we can get rid of this constant offset term. To make the terms in (5) work out, we let $\alpha_k \to 0$, but not too fast. Specifically, we choose a sequence $\{\alpha_k\}$ such that

$$\alpha_k \to 0 \text{ as } k \to \infty, \quad \text{and} \quad \sum_{k=1}^{\infty} \alpha_k = \infty. \tag{6}$$

It is an exercise (but a nontrivial one) to show that under these conditions

$$\frac{\sum_{i=0}^{k-1} \alpha_i^2}{\sum_{i=0}^{k-1} \alpha_i} \to 0 \quad \text{as} \quad k \to \infty.$$

Thus we do get guaranteed convergence of the subgradient method when the stepsizes obey (6).

To get an idea of the tradeoffs involved here, suppose that $\alpha_k = \alpha/(k+1)$. Then for large k, we have the approximations

$$\sum_{i=0}^{k-1} \alpha_i \sim \alpha \log k, \quad \text{and} \quad \sum_{i=0}^{k-1} \alpha_i^2 \sim \text{Const} = \alpha^2\pi^2/6$$

that are good as upper and lower bounds to within constants. In this case, the convergence result (5) becomes

$$f_k^{\text{best}} - f^\star \lesssim \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{\alpha \log k} + \text{Const} \cdot \frac{\alpha L^2}{\log k}.$$

So the convergence is extraordinarily slow – *logarithmic* in k.

You can get much better rates than this (but still not great) by decreasing the stepsize more slowly. Consider now $\alpha_k = \alpha/\sqrt{k+1}$. Then for large k

$$\sum_{i=0}^{k-1} \alpha_i \sim (\alpha+1)\sqrt{k}, \quad \text{and} \quad \sum_{i=0}^{k-1} \alpha_i^2 \sim \alpha^2 \log k,$$

and so

$$f_k^{\text{best}} - f^\star \lesssim \frac{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_2^2}{(\alpha+1)\sqrt{k}} + \text{Const} \cdot \frac{\alpha L^2 \log k}{\sqrt{k}}.$$

This is something like $O(1/\sqrt{k})$ convergence. This means that if we want to guarantee $f_k^{\text{best}} - f^\star \leq \epsilon$, we need $k = O(1/\epsilon^2)$ iterations.

In [**?**, Chapter 3], it is shown that there is no better rate of convergence than $O(1/\sqrt{k})$ that holds uniformly across all problems.

Example. Consider the "ℓ_1 approximation problem"

$$\underset{\boldsymbol{x}\in\mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1.$$

We have already looked at the subdifferential of $\|\boldsymbol{x}\|_1$. Specifically, we showed that $\boldsymbol{u}$ is a subgradient of $\|\boldsymbol{x}\|_1$ at $\boldsymbol{x}$ if it satisfies

$$\begin{aligned} u_n &= \operatorname{sign}(x_n) && \text{if } x_n \neq 0, \\ |u_n| &\leq 1 && \text{if } x_n = 0. \end{aligned}$$

Using what is essentially the same argument we can derive the subdifferential form $f(\boldsymbol{x}) = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1$. First consider a vector $\boldsymbol{z}$ that satisfies

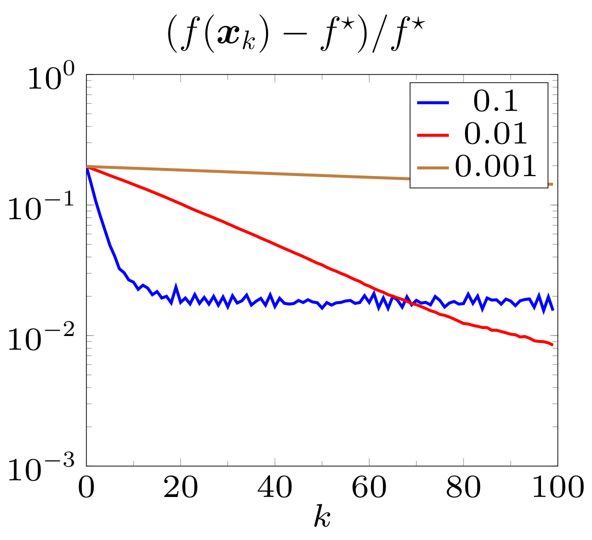
$$\begin{aligned} z_m &= \operatorname{sign}(\boldsymbol{a}_m^{\mathrm{T}}\boldsymbol{x} - b_m) && \text{if } \boldsymbol{a}_m^{\mathrm{T}}\boldsymbol{x} - b_m \neq 0, \\ |z_m| &\leq 1 && \text{if } \boldsymbol{a}_m^{\mathrm{T}}\boldsymbol{x} - b_m = 0. \end{aligned}$$

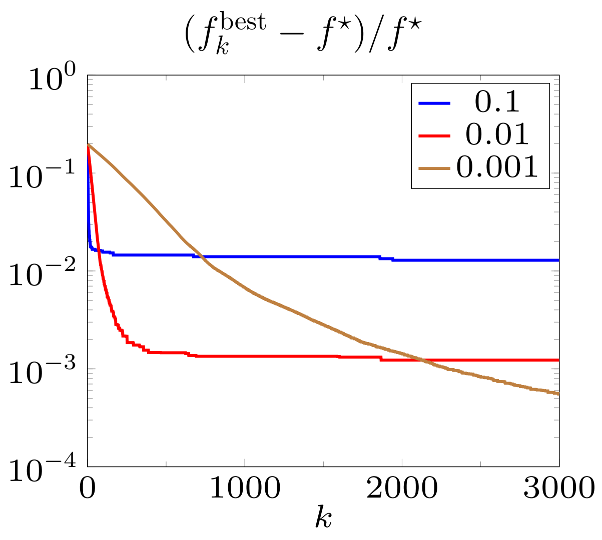
Now consider the vector $\boldsymbol{u} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{z}$. Note that

$$\begin{aligned} \boldsymbol{u}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x}) &= \boldsymbol{z}^{\mathrm{T}}\boldsymbol{A}(\boldsymbol{y} - \boldsymbol{x}) \\ &= \boldsymbol{z}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b} + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) \\ &= \boldsymbol{z}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}) - \boldsymbol{z}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\ &= \boldsymbol{z}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}) - \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1 \\ &\leq \|\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}\|_1 - \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1. \end{aligned}$$

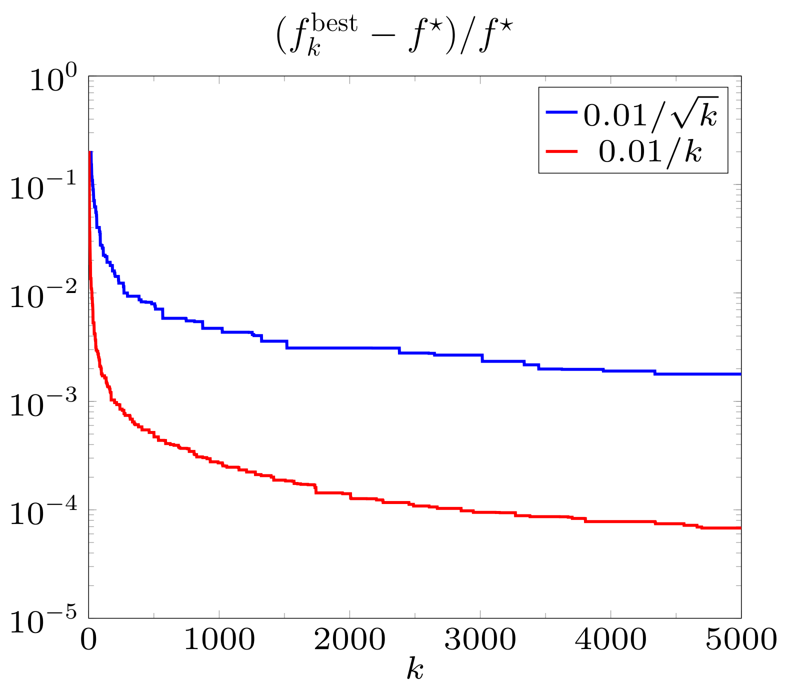
Rearranging this shows that $\boldsymbol{u}$ is a subgradient of $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1$. Using this we can construct a subgradient at each step $\boldsymbol{x}_k$.

Below we illustrate the performance of this approach for a randomly generated example with $\boldsymbol{A} \in \mathbb{R}^{500\times 100}$ and $\boldsymbol{b} \in \mathbb{R}^{1000}$. For three different sizes of fixed step length, $s = 0.1, 0.01, 0.001$, we make quick progress at the beginning, but then saturate, just as the theory predicts:





Here is a run using two different decreasing step size strategies: $\alpha_k = .01/\sqrt{k}$ and $\alpha_k = .01/k$.



As you can see, even though the theoretical worst case bound makes a stepsize of $\sim 1/\sqrt{k}$ look better, in this particular case, a stepsize $\sim 1/k$ actually performs better.

Qualitatively, the takeaways for the subgradient method are:

1. It is a natural extension of the gradient descent formulation
2. In general, it does not converge for fixed stepsizes.
3. If the stepsizes decrease, you can guarantee convergence.
4. Theoretical convergence rates are slow.
5. Convergence rates in practice are also very slow, but depend a lot on the particular example.