Proximal and First-Order Methods for Convex Optimization

John C. Duchi Yoram Singer

January 22, 2013

Abstract

We describe the proximal method for minimization of convex functions. We review classical results, recent extensions, and interpretations of the proximal method that work in online and stochastic optimization settings.

1 Introduction

In this note we describe several first-order algorithms for solving general convex optimization problems. Let $\mathcal{X} \subset \mathbb{R}^n$ be a convex compact subset of $\mathbb{R}^n$. We focus on the following optimization problem

$$\min_{x \in \mathcal{X}} \; f(x), \tag{1}$$

where $f : \mathcal{X} \to \mathbb{R}$ is a convex function. In this general from, finding a good method for solving the problem (1) seems impossible. Nonetheless, by imposing some restrictions we may make progress regardless.

Standard convex programming methodologies developed in the last forty years or so have resulted in extremely fast methods for solving optimization problems. Most methods for solving convex optimization problems are measured by the amount of time or number of iterations required of them to give an ϵ-optimal solution to the problem (1), that is, how long it takes to find some $\widehat{x}$ such that $f(\widehat{x}) - f(x^*) \leq \epsilon$ for an optimal x^*. Modern second order methods [6, 2] are quite efficient, and using just $\mathcal{O}(\log \frac{1}{\epsilon})$ iterations can achieve optimization error ϵ. However, for large scale problems, the time complexity of standard second-order methods can be prohibitive, scaling *at best* as $\mathcal{O}(n^3)$. When n is large, say $n \approx 10^9$, this is clearly infeasible. In such large scale problems, it is reasonable to expect that our representation of problem data is inexact at best. In statistical machine learning problems, for example, this is often the case; generally, many applications do not require accuracy higher than, say $\epsilon = 10^{-2}$, in which case faster but less exact methods become much more desirable.

It is with this motivation that we attack solving the problem (1) in this document, showing several classical (and some more recent) first-order algorithms. These algorithms have the advantage that their iteration costs are low—$\mathcal{O}(n)$ or smaller for n-dimensional problems—but they achieve low accuracy solutions to (1) quickly.

2 Gradient descent and the proximal point algorithm

Let us consider a method for solving problem (1) that is based on repeatedly making simple update steps using linear approximations to the function. Let us assume for now that f is differentiable. In this case,

$$f(y) \approx f(x) + \langle \nabla f(x), y - x \rangle$$

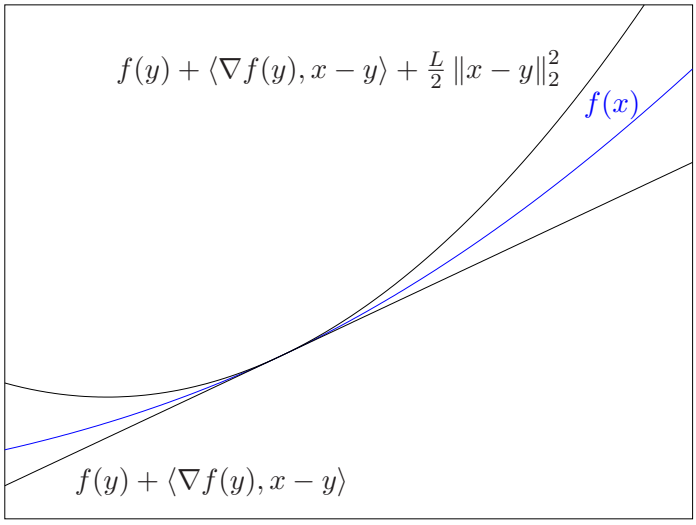

Figure 1: Linear approximation and quadratic upper bound for the function $f(x) = \log(1 + e^x)$ at the point $y = -.5$. This is the upper-bound and approximation of the linear proximal (gradient-based) method (2).

for y relatively close to x. By making this approximation to the function f at a point x, while regularizing via a stepsize $\alpha_k \in \mathbb{R}_+$ to make sure our iterates x_k do not move too far from the previous point, we arrive at the method

$$x_{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2\alpha_k} \|x - x_k\|_2^2 \right\}. \tag{2}$$

2.1 Interpretation

There are several possible interpretations of the above method. The two we favor depend on whether we consider the linearized proximal point method (2) or the projected gradient method, upon which we touch later (they are equivalent). We focus on (2). Let us assume that the function f is smooth, that is, it has L-Lipschitz continuous gradient:

$$\|\nabla f(x) - \nabla f(y)\|_2 \le L \|x - y\|_2 .$$

Then a simple argument with the fundamental theorem of calculus shows that

$$f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|x - y\|_2^2 .$$

Now, consider the linearized proximal minimization step (2). If $\alpha_k \le L^{-1}$, then the proximal step is minimizing an upper bound on the function f that is tight at x_k. Intuitively, then, if we make α_k as large as possible—making the upper bound as tight as possible—while guaranteeing that x_{k+1} still minimizes an upper bound on f, the method should have good performance. See Figure 1 for a visual depiction capturing this intuition for the function $f(x) = \log(1 + e^x)$.

The following theorem describes the convergence rate of the method (2).

Theorem 1. *Let x_k be generated by the gradient method (2) with fixed step size $\alpha \le L^{-1}$. Then for any k and $x^* \in \mathcal{X}$,*

$$f(x_{k+1}) - f(x^*) \le \frac{1}{2k\alpha} \|x_1 - x^*\|_2^2 .$$

The rough take-home convergence rate of the above method is as follows: we simply set $\alpha = L^{-1}$ and let R be an upper bound on the radius of the space $\mathcal{X}$, i.e. $\|x_1 - x^*\|_2^2 \le R^2$. Then

$$f(x_{k+1}) - f(x^*) \le \frac{LR^2}{2k}$$

for any $k \in \mathbb{N}$.

2.2 Stochastic Gradient, Online Learning, and Empirical Losses

In most machine learning problems, the function $f(x)$ takes a particular form, that of an empirical loss. In this case, we assume we have N pieces of data $\{a_i\}_{i=1}^N$, and associated with each datum a_i is an instance-based loss function $f_i(x) = F(x; a_i)$. The objective in problem (1) becomes

$$f(x) = \frac{1}{N} \sum_{i=1}^N f_i(x) = \frac{1}{N} \sum_{i=1}^N F(x; a_i). \tag{3}$$

As a standard example, we often have the data $a_i \in \mathbb{R}^n$, which corresponds to a feature vector, and the instantaneous loss function of the parameters $x \in \mathbb{R}^n$ is

$$F(x; a_i) = \log\left(1 + \exp(\langle a_i, x \rangle)\right),$$

the logistic loss.

Now we have two problems: both n and N may be large, so that computing the gradient $\nabla f(x)$ is quite expensive. In this case, we can instead use a completely random version of the gradient that, in expectation, is correct, but in general will not necessarily be close to $\nabla f(x)$. In this case, we modify the method (2) so that at iteration k, we choose an index i uniformly at random from N, then we simply set $g_k = \nabla F(x_k; a_i) = \nabla f_i(x_k)$, and we replace $\nabla f(x_k)$ with g_k in the update (2). That is, we have the method

$$\begin{aligned} &\text{choose } i \in \{1, \ldots, N\} \text{ at random}, \quad \text{set } g_k = \nabla F(x_k; a_i), \\ &\text{set } x_{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \langle g_k, x \rangle + \frac{1}{2\alpha_k} \|x - x_k\|_2^2 \right\}. \end{aligned} \tag{4}$$

Note that we have the unbiasedness $\mathbb{E}[g_k \mid x_k] = \nabla f(x_k)$, since the index is random.

Intuitively, the method (4) goes in random directions that, after taking enough steps, approximate having actually gone in the correct gradient direction. But because g_k is noisy, instead of using a fixed step size α_k, we usually set α_k to *decrease* with time. This has the effect of increasing the penalty for moving far away from x_k, and the extra penalty allows us to essentially damp out the noise in the gradient estimates. For a typical execution of the stochastic gradient method (4), see Figure 2, where we plot the level curves of the function being minimized and the path taken by the points of stochastic gradient descent (4).

There is nothing in the actual method (4) that forces us to actually have an empirical objective of the form (3), so the method (4) also applies in online learning settings. In this case, instead of the index i being chosen at random, indices i simply arrive in a stream, and the update (4) is applied.

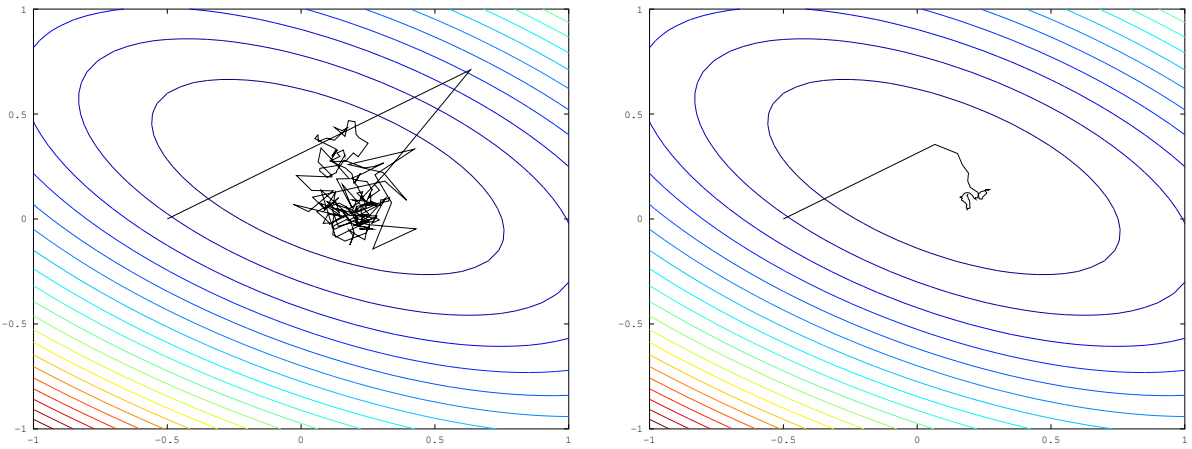

Figure 2: Left: typical execution of the stochastic gradient method (4). Right: typical execution of the stochastic gradient method (4), except that all iterates are averaged (i.e. $\widehat{x}_k = \frac{1}{K}\sum_{k=1}^{K} x_k$)

In many online and stochastic optimization problems the individual iterates x_k are not robust—they are too unstable for effective use. As a consequence, one often averages the iterates over time, as shown on the right side of Figure 2. Averaging gives substantially more robustness while (usually) maintaining optimal convergence guarantees, and a typical result on the convergence of the stochastic gradient method is as follows. Suppose that $\|x_1 - x^*\|_2 \leq R$ as before, but now assume that for each iteration k of the stochastic gradient method, we have $\|g_k\|_2 \leq L$. Then we have the following convergence theorem.

Theorem 2. *Take the stepsize sequence $\alpha_k \equiv R/(L\sqrt{K})$ and define the average $\widehat{x}_k = \frac{1}{K}\sum_{k=1}^{K} x_k$. Then*

$$\mathbb{E}\left[f(\widehat{x}_k)\right] - f(x^*) \leq \frac{2RL}{\sqrt{K}}.$$

This is not as good as the non-stochastic bound from Theorem 1, but it is not improvable without further assumptions.

2.3 Adaptivity of the Quadratic

In some cases, characteristics of the problems arise that make our problem difficult independent of the size of N, the number of examples, or n, the dimensionality of our problem. For example, consider the quadratic function

$$f(x) = \frac{1}{2}\left\langle x, Ax\right\rangle \quad \text{with} \quad A = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}.$$

Starting from the point $x(0) = [.5\ .7]^\top$, the iterations of the standard gradient descent (proximal) method are shown on the right side of Figure 3. Clearly, the minimization of this function is trivial—its minimum is $x = 0$—but if we do not start at the minimizing point, it takes a while to get there.

The reason for the difficulty in this problem is that there is a mismatch between the problem—to minimize the quadratic $f(x) = \frac{1}{2}\left\langle x, Ax\right\rangle$—and the distance measure we use to keep points close in the updates (2). This is, in some sense, the intuition behind the Newton

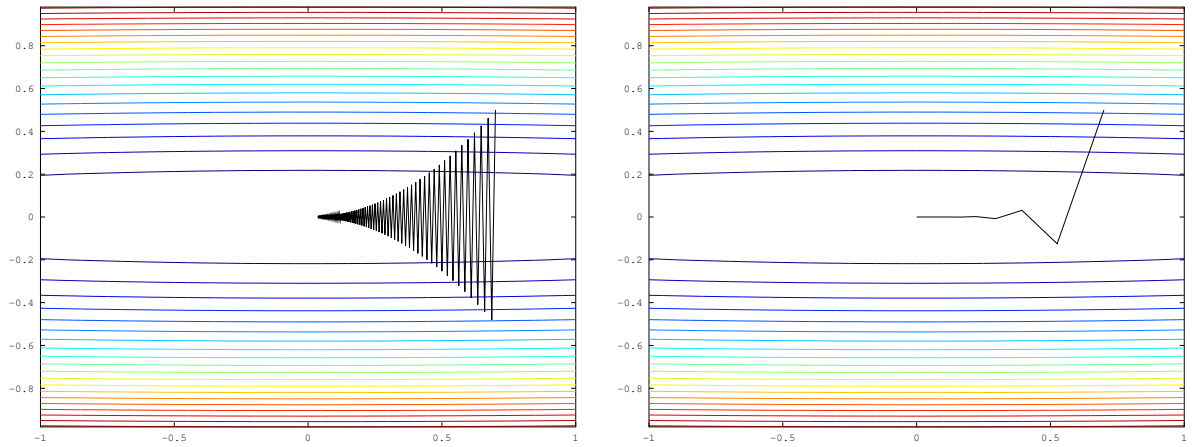

Figure 3: Left: Iterations of the standard proximal gradient method (2) on a poorly conditioned problem. Right: iterations of the simple ADAGRAD variant (5) of the proximal method.

method [2], but it also underlies much of the intuition supporting Duchi, Hazan, and Singer's ADAGRAD method [3] and McMahan and Streeter's similar adaptive method [5]. With that in mind, let us imagine that we use matrix-based norms to measure distances rather than the standard Euclidean norm $\|\cdot\|_2$. That is, for a (positive definite) matrix $B \in \mathbb{R}^{n \times n}$ define the norm $\|\cdot\|_B$ by

$$\|x\|_B^2 := \langle x, Bx \rangle = x^\top B x \geq 0.$$

We will only use diagonal matrices B with positive entries, which clearly guarantee $\langle x, Bx \rangle \geq 0$ for any $x \in \mathbb{R}^n$.

Now, imagine that the set $\mathcal{X} = \mathbb{R}^n$, so that the problem is unconstrained, and we replace $\|\cdot\|_2$ with $\|\cdot\|_B$ in the gradient update (2). Then a little computation gives the update

$$x_{k+1} = x_k - \alpha B^{-1} \nabla f(x_k).$$

So the matrix B has the effect of rescaling the gradient of f, which was exactly our problem with the proximal method (2), as exemplified by the left side of Fig. 3. Indeed, suppose we use the matrix

$$B = A \quad \text{so} \quad B^{-1} = \begin{bmatrix} 10^{-2} & 0 \\ 0 & 1 \end{bmatrix}$$

in the update

$$x_{k+1} := \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \langle \nabla f(x_k), x \rangle + \frac{1}{2\alpha} \|x - x_k\|_B^2 \right\}. \tag{5}$$

Then taking $\alpha = 1$, we solve the problem in one step. The method even works when B is not quite correct, which we show on the right side of Figure 3; there we use $B = \begin{bmatrix} 500 & 0 \\ 0 & 1 \end{bmatrix}$.

It is hopefully clear that choosing a good matrix B in the update (5) can substantially improve the standard gradient method (2). Often, however, such a choice is not obvious, and in stochastic settings such as those in Section 2.2, it is highly non-obvious how to choose this matrix. Moreover, in many stochastic settings, we do not even know the true function we are minimizing, since data simply arrives in a stream, so pre-computing a good distance-generating matrix is impossible.

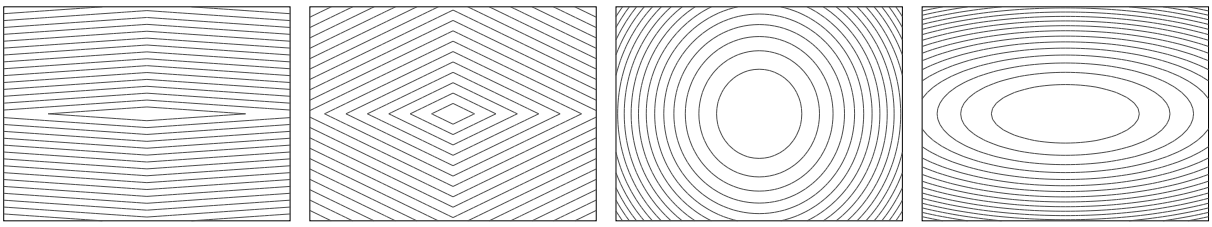

Figure 4: Rescaling effects of ADAGRAD update (7). Far left: contours of original problem. Center left: contours of problem rescaled by ADAGRAD distance. Center right: distance generating function for standard gradient updates (2) and (4). Far right: distance generating function for ADAGRAD update (7).

The standard stochastic gradient method (4) is oblivious to the structure of the problem being solved—it treats all coordinates of the observed gradients equally. Instead of being oblivious, however, we can iteratively develop the matrix B, or, equivalently, the distance we use to keep points close together, by using information on the magnitude of the gradients g_k we observe. Intuitively, by normalizing everything by the magnitude of the gradient coordinates, we move from a problem with level curves that look like those on the right side of Fig. 3 to one in which distances in each coordinate have similar meanings. Another way to gain intuition for ADAGRAD, which we will soon present, is via linear models of the form of the logistic loss $F(x; a_i) = \log(1+\exp(\langle a_i, x\rangle))$. If we view a_i as a feature vector, then when performing updates, we should take more notice of features j that we have *not* seen before than those for which we have substantial information—those we have seen frequently.

Let us now describe our method, using the stochastic setting from Section 2.2 and the update (4) as our starting point. For each k, define G_k to be the diagonal matrix

$$G_k := \operatorname{diag}\left(\sum_{i=1}^{k} g_i g_i^\top\right)^{\frac{1}{2}}. \tag{6}$$

That is, if $g_{i,j}$ denotes the jth entry of the gradient at iteration i, then the jth entry of the diagonal of G_k is $\sqrt{\sum_{i=1}^{k} g_{i,j}^2}$, the root of the sum of squares of the jth coordinate of the gradients g_i. Then for a fixed constant stepsize α, we use the update

$$x_{k+1} = \operatorname*{argmin}_{x\in\mathcal{X}} \left\{\langle g_k, x\rangle + \frac{1}{2\alpha}\left\|x - x_k\right\|_{G_k}^2\right\}. \tag{7}$$

This update is equivalent to the update (4), except that we have replaced the standard ℓ_2-norm $\|\cdot\|_2$ with the adaptive distance $\|\cdot\|_{G_k}$, just as in the update (5).

In Figure 4 we give an example of the rescaling effect that ADAGRAD provides. In particular, we consider minimization of a function of the empirical form (3) with individual terms $F(x; a_i) = \max\{\langle a_i, x\rangle, 0\}$. The level curves of the original function $f(x)$ are presented on the far left side of Fig. 4, and the picture shows that the function is somewhat ill-conditioned. The middle right picture shows the curves of the distance generating function $\|\cdot\|_2^2$ used in the standard gradient and stochastic gradient updates (2) and (4). Note that there is clearly a mismatch between these two. After performing 200 iterations of ADAGRAD, however, using randomly sampled gradients, the scaled distance-generating function used in the update (7) has level curves like those in the rightmost plot of Fig. 4. From the perspective of the method, using this norm to keep points close to one another has the effect of rescaling the level curves of the initial function to look

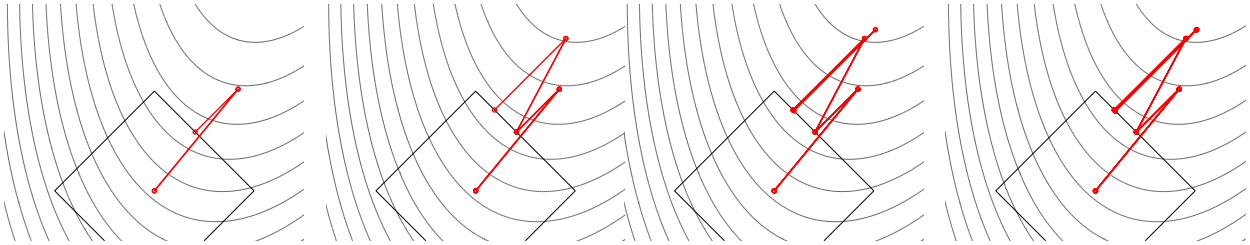

Figure 5: Example execution of the projected gradient method (8), which is equivalent to the proximal point method (2), on minimizing $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ subject to $\|x\|_1 \leq 1$.

like those in the middle right plot of Figure 4, which certainly is better conditioned than the leftmost plot.

In conclusion, we state a theorem describing the convergence of ADAGRAD. The statement of this theorem is more complicated than that of the previous theorems, which only require some smoothness conditions on the function f over $\mathcal{X}$. We need a bit more. First, we assume that $\|x - x^*\|_\infty \leq R_\infty$ for all $x \in \mathcal{X}$, that is, the diameter of $\mathcal{X}$ is bounded in each coordinate. (This is weaker than the ℓ_2-type constraints imposed earlier.) We also define the scalars σ_j^2 to bound the second moments of the jth partial derivative of the functions F, i.e.,

$$\sigma_j^2 \geq \mathbb{E}\left[(\nabla_j F(x; a_i))^2\right] \quad \text{for all } x \in \mathcal{X},$$

where the expectation is taken over the data a_i. Essentially, σ_j measures an average magnitude of the jth coordinate of the gradients $\nabla F(\cdot; a_i)$. Then we have the following theorem.

Theorem 3. *Take the stepsize $\alpha = R_\infty$ and define the average $\widehat{x}_k = \frac{1}{K}\sum_{k=1}^K x_k$. Then*

$$\mathbb{E}[f(\widehat{x}_k)] - f(x^*) \leq \frac{2R_\infty}{K}\sum_{j=1}^n \mathbb{E}\left[\left(\sum_{k=1}^K g_{k,j}^2\right)^{\frac{1}{2}}\right] \leq 2\frac{R_\infty \sum_{j=1}^n \sigma_j}{\sqrt{K}}.$$

Theorem 3 exhibits some of the behavior we hope for from an adaptive method such as ADAGRAD. At a very high level, consider a problem for which we believe the solution vector should be somewhat dense, meaning that all (or most) of the parameters are important. Then measuring convergence via $\|x - x^*\|_\infty$, and the radius R_∞, is (essentially) the best possible way to measure the distance—it does not matter how many parameters we learn. On the data side, at least for linear models, when some features have substantially higher magnitude than others, the difference between $\|g_k\|_2$, which determines the behavior of the standard stochastic method (4), and the sum $\sum_{j=1}^n \sigma_j$ is not substantial.

3 Examples

3.1 Alternate interpretations

For a potential second interpretation, we look at the projected gradient method. With a little algebra, it is clear that the proximal update (2) is equivalent to

$$\begin{aligned} x_{k+1} &= \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \alpha_k \left\langle \nabla f(x_k), x \right\rangle + \frac{1}{2} \left\| x - x_k \right\|_2^2 \right\} \\ &= \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \left\| x - \left[x_k - \alpha_k \nabla f(x_k) \right] \right\|_2^2 \right\}, \end{aligned}$$

or the two step projected gradient iteration

$$\widetilde{x}_{k+1} = x_k - \alpha_k \nabla f(x_k) \quad \text{and} \quad x_{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \left\| x - \widetilde{x}_{k+1} \right\|_2^2 \right\}. \tag{8}$$

In this case, the algorithm simply makes a step in the direction of steepest descent, then projects back onto the feasible set. The points x_k roughly work around the boundary of the convex constraint set $\mathcal{X}$, following the contours of the function f, until the function is minimized. In Figure 5, we plot an example of the performance of the method on minimizing $\|Ax - b\|_2^2$ subject to $\|x\|_1 \le 1$ in two dimensions. The contours of the function are the curved lines, with the global minimum of the function above the upper right corner, while the constraint set is the box. The iterates come in pairs of images, the first in each pair being the step $\widetilde{x}_{k+1} = x_k - \alpha_k \nabla f(x_k)$ outside the constraint set, the second being the projection back into the set $\mathcal{X} = \{x : \|x\|_1 \le 1\}$. We see that the method makes progress by moving along the edge of the set until it arrives at the minimum of f over the constraints.

A Proofs

We prove each of our results using what are known as Lyapunov or potential function arguments. Roughly, in these arguments, one shows that the iterates of the method get closer to (some) optimal point x^* for the problem being solved. The function used to measure this distance is known as the *Lyapunov* function, and one often derives a type of recursive inequality that can be applied to show that this function—or one related to it—goes to zero. This style of proof is classical in optimization and online learning, so we go through the arguments in some detail.

The proofs of each of these results rely on two main observations, which we carefully exhibit in what follows. But we note them here, as they are quite useful in general.

Observation 1. *Let $C \subset \mathbb{R}^n$ be a closed convex set. Then for any point $y \in \mathbb{R}^n$, the projection*

$$\pi_C(y) := \operatorname*{argmin}_{x \in C} \left\| x - y \right\|_2$$

exists, and for any $x \in C$, we have

$$\left\| \pi_C(y) - x \right\|_2 \le \left\| y - x \right\|_2.$$

Moreover, the angle between $y - \pi_C(y)$ and $x - \pi_C(y)$ is oblique, that is,

$$\left\langle y - \pi_C(y), x - \pi_C(y) \right\rangle \le 0 \quad \textit{for all } x \in C.$$

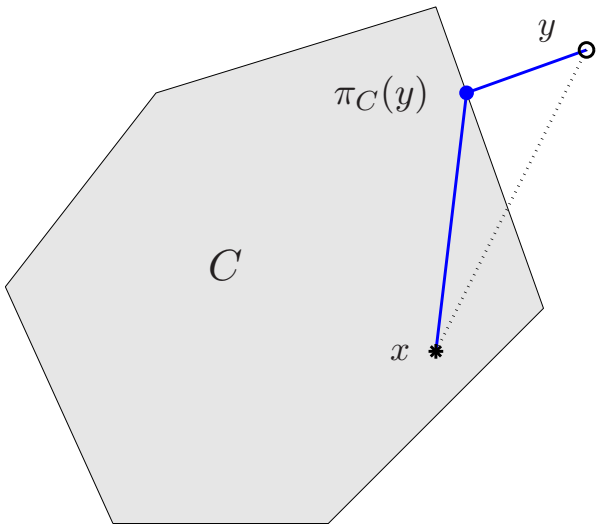

Figure 6: Projection of the point y onto the convex set C. The distance from y to x is greater than distance from $\pi_C(y)$ to x. The angle between $y - \pi_C(y)$ and $x - \pi_C(y)$ is oblique, so $\langle y - \pi_C(y), x - \pi_C(y) \rangle \leq 0$.

For a visual version of this result, see Figure 6. We do not provide a proof here of the observation, though many standard convex optimization textbooks contain proofs [2, 4].

Observation 2. *If f is convex and differentiable, then for any $x, y \in \operatorname{dom} f$, we have*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle . \tag{9}$$

(If f is non-differentiable, then the inequality (9) still holds, but $\nabla f(x)$ is replaced by a vector known as a subgradient.) Additionally, f has L-Lipschitz continuous gradient if and only if

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|x - y\|_2^2 \quad \text{for all } x, y \in \mathcal{X}.$$

Again, we do not provide a proof of this fact, referring back to the linear function that underestimates f in Figure 1 for intuition. (See the books by Boyd and Vandenberghe [2] or Hiriart-Urruty and Lemaréchal [4] for proof of this fact.)

A.1 Proof of Theorem 1

The proof of convergence of the method (2) is one of the standard proofs in convex optimization, and it is worth knowing inside and out. As we will see later, the proof straightforwardly generalizes to to stochastic optimization problems. The idea in this (and in most proofs) is to fix some $x^* \in \mathcal{X}$ to be an "optimal" point, which we use as a comparator to the points x_k the algorithm chooses. In this theorem, our Lyapunov function will be $f(x_{k+1}) - f(x^*)$, which we show goes to zero for any x^*.

The analysis of convergence we give for the method (2) has three main "tricks," which we will be careful to explain in what follows. The first two tricks are to use the convexity of the function f and the L-Lipschitz continuity of ∇f. In particular, the first order convexity inequality tells us that for any $x^* \in \mathcal{X}$,

$$f(x_k) - f(x^*) \leq \langle \nabla f(x_k), x_k - x^* \rangle . \tag{10}$$

Now we use the second trick, that is, the continuity of $\nabla f(x_k)$, to introduce x_{k+1} into the inner product above. The hope here is that since x_{k+1} minimizes the single step (2), we should be able to get some benefit. Indeed, we from the L-Lipschitz continuity of ∇f, we have

$$f(x_{k+1}) \le f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_k - x_{k+1}\|_2^2 .$$

By rearranging the above, we see that

$$\langle \nabla f(x_k), x_k - x_{k+1} \rangle \le f(x_k) - f(x_{k+1}) + \frac{L}{2} \|x_k - x_{k+1}\|_2^2 ,$$

which we combine with the first-order convexity inequality (10) to get

$$\begin{aligned} f(x_k) - f(x^*) &\le \langle \nabla f(x_k), x_k - x^* \rangle \\ &= \langle \nabla f(x_k), x_{k+1} - x^* \rangle + \langle \nabla f(x_k), x_k - x_{k+1} \rangle \\ &\le \langle \nabla f(x_k), x_{k+1} - x^* \rangle + \frac{L}{2} \|x_k - x_{k+1}\|_2^2 + f(x_k) - f(x_{k+1}). \end{aligned}$$

Moving the $f(x_k) - f(x_{k+1})$ terms, we get the important intermediate result from our two tricks of convexity and smoothness:

$$f(x_{k+1}) - f(x^*) \le \langle \nabla f(x_k), x_{k+1} - x^* \rangle + \frac{L}{2} \|x_k - x_{k+1}\|_2^2 . \tag{11}$$

The bound (11) shows that our Lyapunov function in this case is $f(x_{k+1}) - f(x^*)$.

Now we use the third trick, which relies on the fact that x_{k+1} is the projection of

$$\widetilde{x}_{k+1} := x_k - \alpha_k \nabla f(x_k)$$

onto the set $\mathcal{X}$. In particular, recalling Observation 1, we have that for any $y \in \mathcal{X}$,

$$\langle x_k - \alpha_k \nabla f(x_k) - x_{k+1}, y - x_{k+1} \rangle \le 0,$$

or, rewriting slightly,

$$\langle \nabla f(x_k), x_{k+1} - y \rangle \le \frac{1}{\alpha_k} \langle x_{k+1} - x_k, y - x_{k+1} \rangle . \tag{12}$$

By choosing $y = x^*$ in the inequality (12) above, we rearrange and see that

$$\langle \nabla f(x_k), x_{k+1} - x^* \rangle \le \frac{1}{\alpha_k} \langle x_{k+1} - x_k, x^* - x_{k+1} \rangle .$$

By some simple algebra, we see that

$$\langle x_{k+1} - x_k, x^* - x_{k+1} \rangle = \frac{1}{2} \|x_k - x^*\|_2^2 - \frac{1}{2} \|x_{k+1} - x^*\|_2^2 - \frac{1}{2} \|x_{k+1} - x_k\|_2^2 .$$

In particular, the first-order optimality condition (12) for the update (2) implies that

$$\begin{aligned} &\langle \nabla f(x_k), x_{k+1} - x^* \rangle \\ &\le \frac{1}{2\alpha_k} \left[\|x_k - x^*\|_2^2 - \|x_{k+1} - x^*\|_2^2 - \|x_{k+1} - x_k\|_2^2 \right] . \end{aligned} \tag{13}$$

Now we are off to the races. We will use the negative $\|x_{k+1} - x_k\|_2^2$ term in the inequality (13) to cancel the positive $\frac{L}{2}\|x_k - x_{k+1}\|_2^2$ term in the intermediate inequality (11), and the rest of the terms will telescope. Indeed, to cancel appropriately, all we need is that $\alpha_k \le L^{-1}$. In this case, we combine (13) with the bound (11) and have

$$\begin{aligned}
&f(x_{k+1}) - f(x^*) \\
&\le \frac{1}{2\alpha_k}\left[\|x_k - x^*\|_2^2 - \|x_{k+1} - x^*\|_2^2 - \|x_{k+1} - x_k\|_2^2\right] + \frac{L}{2}\|x_k - x_{k+1}\|_2^2 \\
&\le \frac{1}{2\alpha_k}\left[\|x_k - x^*\|_2^2 - \|x_{k+1} - x^*\|_2^2\right].
\end{aligned}$$

For now, let us assume that $\alpha_k \equiv \alpha$, so that it is a constant. By summing the inequality, we have

$$\sum_{k=1}^{K} f(x_{k+1}) - f(x^*) \le \frac{1}{2\alpha}\sum_{k=1}^{K}\|x_k - x^*\|_2^2 - \|x_{k+1}\|_2^2 \le \frac{1}{2\alpha}\|x_1 - x^*\|_2^2. \tag{14}$$

We have proved the following theorem.

By using the nearly final bound (14), we see that

$$K \min_{k\in\{1,\ldots,K\}}\left[f(x_{k+1}) - f(x^*)\right] \le \sum_{k=1}^{K} f(x_{k+1}) - f(x^*) \le \frac{1}{2\alpha}\|x_1 - x^*\|_2^2.$$

Since ∇f is L-Lipschitz continuous and $\alpha^{-1} \ge L$, we know that

$$f(x) \le f(x_k) + \langle \nabla f(x_k), x - x_k\rangle + \frac{1}{2\alpha}\|x - x_k\|_2^2.$$

In particular, x_{k+1} minimizes the right hand side of the above equation (see also Fig. 1), so $f(x_k)$ is non-increasing.

A.2 Proof of Theorem 2

The proof of this theorem is even simpler than that of Theorem 1. The key insight here is that, as described in the update (8), while we may not decrease the objective, we decrease our distance to an optimal point x^*. To that end, we measure our error in terms of the Lyapunov (or potential) function

$$\frac{1}{2}\|x_k - x^*\|_2^2.$$

All we need to do is expand this distance and perform some algebra. Recall the definition (8) of the update for x_{k+1}. Since x_{k+1} is the projection of the point $\widetilde{x}_{k+1}$ onto $\mathcal{X}$, we have (recall Observation 1)

$$\begin{aligned}
\frac{1}{2}\|x_{k+1} - x^*\|_2^2 &\le \frac{1}{2}\|\widetilde{x}_{k+1} - x^*\|_2^2 \\
&= \frac{1}{2}\|x_k - \alpha_k g_k - x^*\|_2^2 \\
&= \frac{1}{2}\|x_k - x^*\|_2^2 + \alpha_k\langle g_k, x^* - x_k\rangle + \frac{\alpha_k^2}{2}\|g_k\|_2^2.
\end{aligned} \tag{15}$$

The bound (15) is our starting point for the remainder of the analysis. We now take expectations. First, we note that

$$\mathbb{E}[\langle g_k, x^* - x_k\rangle] = \mathbb{E}\left[\mathbb{E}[\langle g_k, x^* - x_k\rangle \mid x_k]\right] = \mathbb{E}[\langle \nabla f(x_k), x^* - x_k\rangle]$$

since $\mathbb{E}[g_k \mid x_k] = \nabla f(x_k)$, that is, the gradient is an unbiased estimate of $\nabla f(x_k)$.

Using the first-order convexity inequality (9), we have

$$\langle \nabla f(x_k), x^* - x_k \rangle \le f(x^*) - f(x_k).$$

Applying this to inequality (15), we obtain

$$\frac{1}{2}\mathbb{E}\left[\|x_{k+1} - x^*\|_2^2\right] \le \frac{1}{2}\mathbb{E}\left[\|x_k - x^*\|_2^2\right] + \alpha_k \left(f(x^*) - \mathbb{E}[f(x_k)]\right) + \frac{\alpha_k^2}{2}L^2,$$

since $\|g_k\|_2 \le L$ by assumption. Rewriting this inequality, we obtain

$$\alpha_k \left(\mathbb{E}[f(x_k)] - f(x^*)\right) \le \frac{1}{2}\mathbb{E}\left[\|x_k - x^*\|_2^2\right] - \frac{1}{2}\mathbb{E}\left[\|x_{k+1} - x^*\|_2^2\right] + \frac{\alpha_k^2}{2}L^2,$$

and summing yields

$$\begin{aligned} &\sum_{k=1}^K \alpha_k \left(\mathbb{E}[f(x_k)] - f(x^*)\right) \\ &\qquad \le \frac{1}{2}\mathbb{E}\left[\|x_1 - x^*\|_2^2\right] - \frac{1}{2}\mathbb{E}\left[\|x_{k+1} - x^*\|_2^2\right] + \sum_{k=1}^K \frac{\alpha_k^2}{2}L^2. \end{aligned} \tag{16}$$

The summed bound (16) almost completes the proof. Now, define

$$\widehat{x}_k = \frac{\sum_{k=1}^K \alpha_k x_k}{\sum_{k=1}^K \alpha_k}.$$

This is a convex combination of all the values x observed, whence we obtain

$$\begin{aligned} \mathbb{E}[f(\widehat{x}_k)] - f(x^*) &\le \frac{1}{\sum_{k=1}^K \alpha_k} \sum_{k=1}^K \alpha_k \left(\mathbb{E}[f(x_k)] - f(x^*)\right) \\ &\le \frac{\frac{1}{2}\|x_1 - x^*\|_2^2 + \frac{1}{2}L\sum_{k=1}^K \alpha_k^2}{\sum_{k=1}^K \alpha_k}. \end{aligned}$$

If we take $\alpha_k = \alpha/\sqrt{K}$ for all k, then we have the bound

$$\mathbb{E}[f(\widehat{x}_k)] - f(x^*) \le \frac{\frac{1}{2}\|x_1 - x^*\|_2^2 + \frac{1}{2}L\alpha^2}{\alpha\sqrt{K}} = \frac{\|x_1 - x^*\|_2^2}{2\alpha\sqrt{K}} + \frac{L\alpha}{2\sqrt{K}},$$

which yields Theorem 2. If we take $\alpha_k = \alpha/\sqrt{k}$ for a constant α, we obtain a similar result, except that we get

$$\mathbb{E}[f(\widehat{x}_k)] - f(x^*) = \mathcal{O}\left(\frac{\log K}{\sqrt{K}}\right).$$

A.3 Proof of Theorem 3

The proof of Theorem 3 is similar to that of the previous proof, but we use an adaptive Lyapunov function. We also require the following result, which is analogous to Observation 1, but applies in to slightly different Euclidean-type norms.

Observation 3. *Let $C \subset \mathbb{R}^n$ be a closed convex set and $A \succ 0$ be a positive definite matrix. Then for any point $y \in \mathbb{R}^n$, the projection*

$$\pi_C^A(y) := \operatorname*{argmin}_{x \in C} \|x - y\|_A^2$$

exists, and for any $x \in C$, we have

$$\left\|\pi_C^A(y) - x\right\|_A^2 \le \|y - x\|_A^2 .$$

Given Observation 1, this should come as no surprise.

Now, let us recall the ADAGRAD iteration (7). As we did with the proximal method, we can rewrite this iteration to be a projected gradient scheme. Indeed, we have

$$\widetilde{x}_{k+1} = x_k - G_k^{-1} g_k \quad \text{and} \quad x_{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}} \left\{ \|\widetilde{x}_{k+1} - x\|_{G_k} \right\}. \tag{17}$$

Using the recursion (17), we can apply a very similar Lyapunov analysis to what we did for the standard stochastic gradient method in Theorem 2.

As before, we begin with our Lyapunov function: fixing $x^* \in \mathcal{X}$, we use

$$\frac{1}{2} \|x_{k+1} - x^*\|_{G_k}^2 .$$

By using Observation 3, we find that

$$\begin{aligned}
\frac{1}{2} \|x_{k+1} - x^*\|_{G_k}^2 &\le \frac{1}{2} \|\widetilde{x}_{k+1} - x^*\|_{G_k}^2 \\
&= \frac{1}{2} \left\|x_k - \alpha G_k^{-1} g_k - x^*\right\|_{G_k}^2 \\
&= \frac{1}{2} \|x_k - x^*\|_{G_k}^2 + \alpha \left\langle g_k, x^* - x_k \right\rangle + \frac{\alpha^2}{2} \left\|G_k^{-1} g_k\right\|_{G_k}^2 \\
&= \frac{1}{2} \|x_k - x^*\|_{G_k}^2 + \alpha \left\langle g_k, x^* - x_k \right\rangle + \frac{\alpha^2}{2} \|g_k\|_{G_k^{-1}}^2 .
\end{aligned} \tag{18}$$

Here we have taken advantage of the projected-gradient version (17) of ADAGRAD.

For notational simplicity, let $f_k(x) = F(x; a_k)$ be the stochastic function sampled at iteration k of the ADAGRAD (or stochastic gradient) procedure. Then we apply Observation 2, as in the proof of Theorem 2, to find that

$$\langle g_k, x^* - x_k \rangle \le f_k(x^*) - f_k(x_k).$$

Rewriting this, we have via the iterative bound (18) that

$$\frac{1}{2} \|x_{k+1} - x^*\|_{G_k}^2 \le \frac{1}{2} \|x_k - x^*\|_{G_k}^2 + \alpha \left[f_k(x^*) - f_k(x_k)\right] + \frac{\alpha^2}{2} \|g_k\|_{G_k^{-1}}^2 ,$$

or (dividing by α) that

$$f_k(x_k) - f_k(x^*) \le \frac{1}{2\alpha} \left[\|x_k - x^*\|_{G_k}^2 - \|x_{k+1} - x^*\|_{G_k}^2\right] + \frac{\alpha}{2} \|g_k\|_{G_k^{-1}}^2 .$$

It remains to turn this bound into something useable; to do so, we sum the bound. We thus have

$$\sum_{k=1}^K f_k(x_k) - f_k(x^*) \le \frac{1}{2\alpha} \sum_{k=1}^K \left[\|x_k - x^*\|_{G_k}^2 - \|x_{k+1} - x^*\|_{G_k}^2\right] + \frac{\alpha}{2} \sum_{k=1}^K \|g_k\|_{G_k^{-1}}^2 . \tag{19}$$

On the face of it, it appears that the sum (19) should telescope and provide us the convergence guarantees we desire, but this is not quite the case, as the norm $\|\cdot\|_{G_k}$ changes.

We now show how to turn the bound (19) into a convergence guarantee. To do so, we note two important results: first, we have the inequality $G_{k+1} \succeq G_k$ for all k, meaning that $\langle x, G_{k+1}x\rangle \geq \langle x, G_k x\rangle$ for all $k \in \mathbb{N}$, by the definition (6) of G_k. Secondly, we have the following useful lemma [1, 3, 5].

Lemma 4. *Let G_k be defined by the update* (6)*. For any sequence of vectors $\{g_k\} \subset \mathbb{R}^n$, the inequality*

$$\sum_{k=1}^{K} \left\langle g_k, G_k^{-1} g_k \right\rangle \leq 2 \sum_{j=1}^{n} \sqrt{\sum_{k=1}^{K} g_{k,j}^2}.$$

Taking Lemma 4 as a given for now, we give the remainder of the proof. It remains to control the two sums in the inequality (19).

Starting with the first, we have

$$\begin{aligned}
&\sum_{k=1}^{K} \left[\|x_k - x^*\|_{G_k}^2 - \|x_{k+1} - x^*\|_{G_k}^2 \right] \\
&\qquad = \|x_1 - x^*\|_{G_1}^2 - \|x_{K+1} - x^*\|_{G_K}^2 + \sum_{k=2}^{K} \left[\|x_k - x^*\|_{G_k}^2 - \|x_k - x^*\|_{G_{k-1}}^2 \right]
\end{aligned}$$

Looking at the second sum, we note that we can rewrite this as a difference of matrices in the norm, and moreover, since the matrices G_k are diagonal, we have $\|y\|_{G_k}^2 \leq \|\mathrm{diag}(G_k)\|_1 \|y\|_\infty^2$ by a simple calculation, for any $y \in \mathbb{R}^n$. With this in mind, we find that

$$\begin{aligned}
\sum_{k=1}^{K} \left[\|x_k - x^*\|_{G_k}^2 - \|x_{k+1} - x^*\|_{G_k}^2 \right] &= \|x_1 - x^*\|_{G_1}^2 - \|x_{K+1} - x^*\|_{G_K}^2 + \sum_{k=2}^{K} \|x_k - x^*\|_{G_k - G_{k-1}}^2 \\
&\leq \|\mathrm{diag}(G_1)\|_1 \|x_1 - x^*\|_\infty^2 + \sum_{k=2}^{K} \|\mathrm{diag}(G_k - G_{k-1})\|_1 \|x_k - x^*\|_\infty^2 .
\end{aligned}$$

Since the diagonals of G_k are only increasing—recall the definition (6) of the matrices—and since we assumed $\|x_k - x^*\|_\infty \leq R_\infty$, we find

$$\sum_{k=2}^{K} \|\mathrm{diag}(G_k - G_{k-1})\|_1 \|x_k - x^*\|_\infty^2 \leq \sum_{k=2}^{K} \|\mathrm{diag}(G_k - G_{k-1})\|_1 R_\infty^2 = \mathrm{tr}(G_K - G_1) R_\infty^2.$$

Summarizing, we find that

$$\sum_{k=1}^{K} \left[\|x_k - x^*\|_{G_k}^2 - \|x_{k+1} - x^*\|_{G_k}^2 \right] \leq R_\infty^2 \,\mathrm{tr}(G_K) = R_\infty^2 \sum_{j=1}^{n} \sqrt{\sum_{k=1}^{K} g_{k,j}^2}. \tag{20}$$

By combining the "telescoping" bound (20) with Lemma 4 and our original inequality (19), we arrive at the inequality

$$\sum_{k=1}^{K} f_k(x_k) - f_k(x^*) \leq \left[\frac{1}{2\alpha} R_\infty^2 + \alpha \right] \sum_{j=1}^{n} \sqrt{\sum_{k=1}^{K} g_{k,j}^2}.$$

Noting as in the proof of Theorem 2 that $\mathbb{E}[f_k(x_k) \mid x_k] = f(x_k)$, we can take an average (i.e. set $\widehat{x}_K = \frac{1}{K}\sum_{k=1}^K x_k$) and get

$$\mathbb{E}[f(\widehat{x}_K)] - f(x^*) \le \frac{1}{K}\left[\frac{1}{2\alpha}R_\infty^2 + \alpha\right]\mathbb{E}\left[\sum_{j=1}^n \left(\sum_{k=1}^K g_{k,j}^2\right)^{\frac{1}{2}}\right],$$

which is the statement of the theorem. (The final inequality in the theorem follows from Jensen's inequality.) □

We have not provided the proof of Lemma 4, which we now do. Note that the coordinates in the lemma are all updated independently, so we can instead prove a bound on a scalar sequence, then apply the bound to each coordinate to obtain the lemma. We adapt the proof of Lemma 4 of Duchi et al. [3]. Let $\{a_1, a_2, \ldots, a_K\} \subset \mathbb{R}$ be a sequence of real numbers. We show that

$$\sum_{k=1}^K \frac{a_k^2}{\sqrt{\sum_{i=1}^k a_i^2}} \le 2\left(\sum_{k=1}^K a_k^2\right)^{\frac{1}{2}}. \tag{21}$$

The inequality clearly holds for $k = 1$, so we may assume that it holds for $K - 1$. In this case, we have

$$\sum_{k=1}^K \frac{a_k^2}{\sqrt{\sum_{i=1}^k a_i^2}} \le 2\left(\sum_{k=1}^{K-1} a_k^2\right)^{\frac{1}{2}} + \frac{a_K^2}{\sqrt{\sum_{k=1}^K a_k^2}}.$$

Now, by the concavity of $\sqrt{\cdot}$, we have the inequality $\sqrt{b-a} \le \sqrt{b} - a/2\sqrt{b}$ for a, b such that $b - a \ge 0$ (this is the first order convexity inequality (9) reversed). Applying this concavity inequality, we see that

$$\begin{aligned}
2\left(\sum_{k=1}^{K-1} a_k^2\right)^{\frac{1}{2}} + \frac{a_K^2}{\sqrt{\sum_{k=1}^K a_k^2}} &= 2\left(\sum_{k=1}^K a_k^2 - a_K^2\right)^{\frac{1}{2}} + \frac{a_K^2}{\sqrt{\sum_{k=1}^K a_k^2}} \\
&\le 2\left(\sum_{k=1}^K a_k^2\right)^{\frac{1}{2}} - \frac{a_K^2}{\sqrt{\sum_{k=1}^K a_k^2}} + \frac{a_K^2}{\sqrt{\sum_{k=1}^K a_k^2}} = 2\left(\sum_{k=1}^K a_k^2\right)^{\frac{1}{2}}.
\end{aligned}$$

Thus the inequality (21) holds, as does the lemma.

References

[1] P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[4] J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I & II*. Springer, 1996.

[5] B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. In *Proceedings of the Twenty Third Annual Conference on Computational Learning Theory*, 2010.

[6] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics, 1994.