Course Notes on Acceleration and Hedging in Optimization

Short Course given by Jason Altschuler
Princeton ML Theory Summer School 2024
Notes taken by Alexander Terenin

1 Setting the Stage

A typical machine learning question is to find a model that fits our data, which involves optimization. The canonical algorithm is gradient descent. Our theme will be non-greedy approaches to algorithm design. We will study the question how to analyze optimization, if not greedy, focused on the simplest algorithms such as gradient descent in simplest settings, such as smooth convex optimization.

We will study Chebyshev step sizes, momentum, and random step sizes for quadratics, where one can use linear structure to understand behavior. We will then look at arbitrary first-order algorithms and study automated ways to study convergence, via semi-definite programs and similar. We will conclude by studying silver step sizes.

2 Foundational Questions about Gradient Descent

Let's study why gradient descent moves in the $-\nabla f$ direction, with the aim of answering whether it converges and for what choices of step sizes, and, if so, how quickly this occurs. We start with the former. Our optimization problem and algorithm is

$$\min_x f(x) \qquad\qquad x_{t+1} = x_t - \alpha_t \nabla f(x_t) \tag{1}$$

Locally, for $\|v\| \leq \delta$ with δ small, we have

$$f(x+v) \approx f(x) + \langle \nabla f(x), v \rangle. \tag{2}$$

What's the best direction for this? This is

$$\arg\min_{\|v\| \leq \delta} f(x+v) \tag{3}$$

which is intractable, so we localize, to obtain

$$\arg\min_{\|v\| \leq \delta} f(x) + \langle \nabla f(x), v \rangle \tag{4}$$

which one can solve to obtain

$$v^* = -\delta \frac{\nabla f(x)}{\|\nabla f(x)\|}. \tag{5}$$

From this, we can make precise how much progress we make at every step. We will later see how to do this automatically on a computer to obtain precise rates. For now, however, we start with quadratics. Let

$$f(x) = \frac{1}{2} x^T H x - b^T x. \tag{6}$$

We will assume spectral bounds on H, namely

$$mI \preceq H \preceq MI \tag{7}$$

which roughly correspond to strong convexity and smoothness. This in particular means we can bound f from both sides by quadratics $\frac{m}{2}x^2$ and $\frac{M}{2}x^2$. The motivation here is least squares regression, namely

$$Ax = v \tag{8}$$

which can be written in optimization form as

$$\min_x \|Ax - v\|^2 \qquad \|Ax - v\|^2 = \frac{1}{2} x^T A^T A x - v^T A x + c \tag{9}$$

for an appropriate c. We can get the optimum via the stationarity conditions

$$0 = \nabla f(x) = Hx - b \tag{10}$$

which gives

$$x^* = H^{-1} b. \tag{11}$$

Gradient descent is

$$x_{t+1} = x_t - \alpha_t \nabla f(x_t) = x_t - \alpha_t (H x_t - b). \tag{12}$$

By adding and subtracting, we get

$$x_{t+1} - x^* = x_t - x^* - \alpha_t H_t (x_t - x^*) \tag{13}$$

which means that

$$x_{t+1} - x^* = (I - \alpha_t H_t)(x_t - x^*). \tag{14}$$

With this re-centered form, we can now ask all the preceding questions about gradient descent. This is a linear dynamical system, possibly time-invariant. Does it converge? Consider constant step size α_t. We aks whether there exists an α such that

$$\max_{mI \preceq H \preceq MI} \rho(I - \alpha H) < 1. \tag{15}$$

This amounts to considering the eigenvalues of $I - \alpha H$ are $1 - \alpha\lambda$, where λ ranges in $[m, M]$.

Theorem 1. *If one chooses sufficiently-small constant step sizes, gradient descent converges.*

We now ask about rates. Let

$$R_n = \inf_{\alpha_0,..,\alpha_{n-1}} \sup_{\substack{mI \le H \le MI \\ x_0 \neq x^*}} \frac{\|x_n - x^*\|}{\|x_0 - x^*\|}. \tag{16}$$

Understanding this quantity lets us understand non-adaptive step sizes, but which depend on the problem's parameters. For $n = 1$, we have

$$R_1 = \inf_{\alpha} \sup_{\substack{mI \le H \le MI \\ x_0 \neq x^*}} \frac{\|x_1 - x^*\|}{\|x_0 - x^*\|} \tag{17}$$

$$\overset{\text{(i)}}{=} \inf_{\alpha} \sup_{\substack{mI \le H \le MI \\ x_0 \neq x^*}} \frac{\|(I - \alpha H)(x_0 - x^*)\|}{\|x_0 - x^*\|} \tag{18}$$

$$\overset{\text{(ii)}}{=} \inf_{\alpha} \sup_{mI \le H \le MI} \|I - \alpha H\| \tag{19}$$

$$\overset{\text{(iii)}}{=} \inf_{\alpha} \sup_{\lambda \in [m,M]} |1 - \alpha\lambda| \tag{20}$$

$$\overset{\text{(iv)}}{=} \frac{M - m}{M + m} \tag{21}$$

where (i) is by the gradient descent step identity, (ii) is by definition of an operator norm, (iii) is by the eigenvalue characterization of operator norms, and (iv) is as follows. One can see that the resulting min-max problem is easy to solve by drawing a picture, and gives $\alpha^* = \frac{2}{M-m}$ with optimal λ^* on the boundary of the interval: plugging this in gives the contraction value. One can iterate this to show

$$\|x_n - x^*\| \le R_1^n \|x_0 - x^*\| \tag{22}$$

where we have assumed constant step size. From this, we obtain the following.

Theorem 2. *By carefully choosing a constant step size, gradient descent converges at a rate of*

$$\frac{M-m}{M+m} < 1. \tag{23}$$

This is called both a linear rate, and an exponential rate, depending on the context. Let us now consider two different step sizes. Now, we have

$$R_2 = \inf_{\alpha,\beta} \sup_{\substack{mI \leq H \leq MI \\ x_0 \neq x^*}} \frac{\|(I-\beta H)(I-\alpha H)(x_0 - x^*)\|}{\|x_0 - x^*\|} \tag{24}$$

$$= \inf_{\alpha,\beta} \|(I-\beta H)(I-\alpha H)\| \tag{25}$$

$$= \inf_{\alpha,\beta} \sup_{\lambda \in [m,M]} |(1-\alpha\lambda)(1-\beta\lambda)|. \tag{26}$$

Thus, asking about step sizes α, β is equivalent to asking about a polynomial p of degree 2 such that $p(0) = 1$, where

$$p(\lambda) = (1-\alpha\lambda)(1-\beta\lambda). \tag{27}$$

The constraint $p(0) = 1$ arises by multiplying out the polynomial, which shows that at $\lambda = 0$ the polynomial is equal to one, but is otherwise not fixed. Hence

$$R_2 = \inf_{\substack{\deg(p)=2 \\ p(0)=1}} \sup_{\lambda \in [m,M]} |p(\lambda)|. \tag{28}$$

We can plot this polynomial to see that we get better rates by varying step sizes. There are two different roots. The roots are different than R_1. Thus, $\sqrt{R_2} < R_1$.

Theorem 3. *By carefully choosing a time-varying pair of step sizes we cycle through, gradient descent converges at a rate of*

$$\sqrt{R_2} < \frac{M-m}{M+m} < 1. \tag{29}$$

Let's try to handle the general case. By a similar argument to previously, we obtain a min-max problem over a space of polynomials, namely

$$R_n = \inf_{\substack{\deg(p_n)=n \\ p(0)=1}} \sup_{mI \leq H \leq MI} \|p_n(H)\| = \inf_{\substack{\deg(p_n)=n \\ p(0)=1}} \sup_{\lambda \in [m,M]} \|p_n(\lambda)\| \tag{30}$$

where $p_n(H)$ is understood in the sense of matrix powers. We get the following:

1. The rates are better for larger n.
2. Each step size is different.

3. The optimal step-size equalizes a set of worst-case functions.

The optimal polynomial is

$$p_n(\lambda) = \frac{T_n(L(\lambda))}{T_n(L(0))} \tag{31}$$

where $L : [-1, 1] \to [m, M]$ linearly without reflections and T_n are Chebyshev polynomials. The optimal step sizes are explicit functions of the inverse roots. This gives the following result.

Theorem 4. *By carefully choosing a step size schedule, gradient descent converges at a rate of*

$$R_n^{1/n} \approx \frac{\sqrt{M} - \sqrt{m}}{\sqrt{M} + \sqrt{m}} \tag{32}$$

where the result holds in an asymptotic limit with small quantifiable finite-n error.

For $n = 1$, we have $R_1 \approx 1 - \frac{m}{M}$, so the number of iterations for $R_1^n < \varepsilon$ is $\Theta\big(\frac{M}{m} \log \frac{1}{\varepsilon}\big)$. For general n, we have $R_n^{1/n} \approx 1 - \sqrt{\frac{m}{M}}$, so the number of iterations for $R_n < \varepsilon$ is $\Theta\Big(\sqrt{\frac{M}{m}} \log \frac{1}{\varepsilon}\Big)$.

3 Alternative Implementations

3.1. Momemtum. We begin by defining Chevyshev polynomials.

Definition 5. *Define the Chebyshev polynomials $R_n(z)$ by one of the following equivalent definitions:*

1. *Explicit: $T_n(z) = \cos(na\cos(z))$ for $|z| \le 1$.*
2. *Roots: $\big\{\cos\big(\frac{2t+1}{2n}\pi\big)\big\}_{t=0}^{n-1}$.*
3. *Three-term recurrence: $T_0(z) = 1$, $T_1(z) = z$, and $T_{n+1}(z) = 2zT_n(z) - T_{n-1}(z)$.*

One can draw the behavior of the roots by drawing equispaced roots of unity in the complex plane, and shifting them down to the x-axis. From the three-term recurrence, we can see another way to get improved rates, via the heavy-ball method. This is defined as

$$x_{n+1} - x^* = p_{n+1}(H)(x_0 - x^*). \tag{33}$$

One can show for appropriate a, b, c that the polynomial satisfies

$$p_{n+1}(H) = (aH + b)L(H)p_n(H) + cp_{n-1}(H). \tag{34}$$

This means that

$$x_{n+1} - x^* = p_{n+1}(H)(x_0 - x^*) \tag{35}$$
$$= aHp_n(H)(x_0 - x^*) + bp_n(H)(x_0 - x^*) + cp_{n-1}(H)(x_0 - x^*) \tag{36}$$
$$= a\nabla f(x_n) + b(x_n - x^*) + c(x_{n-1} - x^*) \tag{37}$$

which expands into gradient descent with momentum. In particular, this shows that, in some sense, the reason momentum uses two terms is because orthogonal polynomials are always given by three-term recurrences.

3.2. Random step sizes. Suppose that $\frac{1}{\alpha_t} \sim \mu$ are IID. What's the best μ? Does this converge quickly? We again consider

$$\sup_{\substack{mI \le H \le MI \\ x_0 \neq x^*}} \left(\frac{\|x_n - x^*\|}{\|x_0 - x^*\|} \right)^{1/n} = \sup_{\lambda \in [m,M]} \left| \prod_{t=0}^{n-1} (1 - \alpha_t \lambda) \right|^{1/n} \tag{38}$$
$$= \sup_{\lambda \in [m,M]} \exp\left(\frac{1}{n} \sum_{t=1}^{n-1} \log|1 - \alpha_t \lambda| \right) \tag{39}$$
$$\le \exp\left(\frac{1}{n} \sup_{\lambda \in [m,M]} \sum_{t=1}^{n-1} \log|1 - \alpha_t \lambda| \right) \tag{40}$$
$$\approx \exp\left(\sup_{\lambda \in [m,M]} \mathop{\mathbb{E}}_{\alpha \sim \mu} \log|1 - \alpha\lambda| \right) \tag{41}$$

which is permutation-invariant. This shows nothing is lost by passing to the IID case, and gives an idea about how to study limiting behavior.

To proceed, we need to understand the limiting distribution of the roots of Chebyshev polynomials. One can show that this distribution is the arcsine distribution on $[-1, 1]$, with density

$$\frac{1}{\pi\sqrt{(1-z)(1+z)}}. \tag{42}$$

3.3. Connection to potential theory. Consider the following problems.

1. *Stepsize problem*: without knowing about Chebyshev polynomials, one can try to find the distribution for which

$$\inf_{\mu} \sup_{\lambda \in [m,M]} \mathop{\mathbb{E}}_{\beta \sim \mu} \log\left|1 - \frac{\lambda}{\beta}\right|. \tag{43}$$

2. *Electrostatics problem*: we want to find a distribution for which

$$\inf_{\mu \in \mathcal{P}([m,M])} \mathop{\mathbb{E}}_{\beta, \lambda \sim \mu} \log \frac{1}{\beta - \lambda}. \tag{44}$$

For the latter, we now state *Frostman's Theorem*. Define $\phi_M(\lambda) = \mathbb{E}_{\beta\sim\mu} \frac{1}{|\beta-\lambda|}$, so that the electrostatics problem is $\mathbb{E}_{\lambda\sim\mu}\, \phi_M(\lambda)$.

Theorem 6. *For the electrostatics problem, there exists an optimal μ which is characterized by an* EQUALIZING PROPERTY: *$\phi_M(\lambda)$ is constant on $[m, M]$.*

Some of these ideas also apply to the stepsize problem. By equalization, optimality of μ means that in the limit there is no worst case problem.

4 Beyond Quadratics

Every part of the preceding story breaks down in at least some way outside the quadratic case. We start with the univariate case. We study

$$\min_{x\in\mathbb{R}} f(x) \tag{45}$$

which is strongly convex and smooth, namely

$$m \le f''(x) \le M. \tag{46}$$

We focus on what changes. Gradient descent is

$$x_{t+1} = x_t - \alpha_t f'(x_t). \tag{47}$$

Applying the Fundamental Theorem of Calculus gives

$$f'(x_t) = \int_0^1 f''(ux_t + (1-u)x^*)\,\mathrm{d}u(x_t - x^*) \tag{48}$$

Define

$$\lambda_t = \int_0^1 f''(ux_t + (1-u)x^*)\,\mathrm{d}u \tag{49}$$

so that

$$x_{t+1} - x^* = (x_t - x^*) - \alpha_t\lambda_t(x_t - x^*). \tag{50}$$

Repeating the preceding derivation gives

$$\exp\left(\frac{1}{n}\sup_{\lambda\in[m,M]}\sum_{t=1}^{n-1}\log|1-\alpha_t\lambda_t|\right) \tag{51}$$

where λ_t are now time-varying and can depend on history. The question is whether one can apply something like a law of large numbers, but this does not apply directly because the entries are not independent, and not identically distributed.

The key to handling this will be the equalizing property. The claim is that for any $\lambda \in [m, M]$, we have

$$\mathop{\mathbb{E}}_{\alpha} \log |1 - \alpha\lambda| = \log \frac{\sqrt{M} - \sqrt{m}}{\sqrt{M} + \sqrt{m}} \tag{52}$$

and

$$\mathop{\mathbb{E}}_{\alpha_t} (\log |1 - \alpha_t \lambda_t| \mid \text{history}) = \log \frac{\sqrt{M} - \sqrt{m}}{\sqrt{M} + \sqrt{m}} \tag{53}$$

By choosing α_t randomly, one can protect against the worst case. It is an open problem whether or not this result holds without random step sizes. The difficulty is that learning rates from Chebyshev polynomials provably do not work: randomization seems like it should be essential to get limiting behavior.

5 Performance Estimation Problems

We ask what is the best algorithm we can construct that is good in the worst case. The algorithm design problem is

$$\underbrace{\min_{\text{algorithm}}}_{\substack{\text{algorithm} \\ \text{design}}} \underbrace{\max_{\text{problem}} \text{rate}}_{\substack{\text{algorithm} \\ \text{analysis}}} \tag{54}$$

Performance estimation problems try to solve the inner maximization problem by casting it in the language of convex optimization. A good reference on this topic is tutorials by Adrien Taylor.

The challenge beyond quadratics is that there is no simple way to parameterize optimization over worst-case convex functions. We can ask questions like: what are tight rates for gradient descent, Nesterov's algorithm, or heavy ball? Are there situations where heavy ball doesn't converge? How do we know that $\sqrt{R_2} < R_1$? For the final question, the optimization problem is

$$\min_{\substack{f \in \mathcal{F}_{m,M}, x_0 \neq x^* \\ x_1 = x_0 - \alpha \nabla f(x_0) \\ x_2 = x_1 - \beta \nabla f(x_1)}} \frac{\|x_2 - x^*\|}{\|x_0 - x^*\|}. \tag{55}$$

We usually try to prove this by bounding $\|x_2 - x^*\|^2$ by $R_2 \|x_0 - x^*\|^2$ using valid inequalities, where these are chosen by creativity. Performance estimation problems work by writing

$$R_2 \|x_0 - x^*\|^2 - \|x_1 - x^*\|^2 = \sum_i c_i (\text{valid inequalities}) + (\text{square term}) \tag{56}$$

and solving an optimization problem for the c_i-coefficients and square term. This proof system is complete: it is enough to consider a finite set of inequalities, and these can be efficiently searched.

Let's see if this can be related to something more familiar. Consider a polynomial and ask whether it is non-negative. For instance, consider

$$p(x) = x_1^2 - 4x_1x_2 + 10x^2 + 2x_1x_3 - 8x_2x_3 + 10x_3^2 \tag{57}$$

$$= (x_1 - 2x_2 + x_3)^2 + (x_2 - 2x_3)^2 + 5x_2^2 + 5x_3^2. \tag{58}$$

In the first form, it is not obvious whether it is non-negative, but this is easy to see in the second form. Sum of squares decompositions certify non-negativity of polynomials—this represents the rate of our algorithm—by combining and adding valid inequalities. Finding a sum of squares decomposition amounts to convex optimization.

Lemma 7. *A polynomial p is a sum of squares if and only if there exists a matrix A such that*

$$p(x) = \left\langle A, xx^T \right\rangle. \tag{59}$$

Proof. For a vector c, write

$$p(x) = \sum_i \langle c_i, x \rangle^2 = \sum_i \left\langle c_i c_i^T, xx^T \right\rangle. \tag{60}$$

■

In the preceding formulation, part of the issue is we have $f \in \mathcal{F}_{m,M}$ as our domain of minimization, which is infinite-dimensional. The idea is to replace this with a condition which involves whether or not there exists a function which interpolates the data. So, instead of working directly with feasibility $f \in \mathcal{F}_{m,M}$, we ask whether there exists an f for which $\nabla f(x_i) = g_i$, where $x_i = x_0 - \alpha g_0$ and x_i, g_i are variables we optimize over. In the simplest case, we can use the following theorem, due to Rockafellar.

Theorem 8. *The values $\{x_i, g_i, f_i\}_{i \in I}$ are convex-interpolable if and only if*

$$f_i - f_j - \langle g_j, x_i - x_j \rangle \geq 0. \tag{61}$$

Proof sketch. Consider

$$f(x) = \max_i f_i + \langle g_i, x - x_i \rangle. \tag{62}$$

■

This relates the numbers above with existence of convex function. We will use this to remove dependence on the space of functions $\mathcal{F}_{m,M}$, and replace it with dependence on this set of numbers. To do this, we need a generalization of the preceding statement to strongly convex and smooth functions.

Theorem 9. *The values $\{x_i, g_i, f_i\}_{i\in I}$ are $\mathcal{F}_{m,M}$-interpolable if and only if*

$$0 \geq Q_{ij} = 2(M-m)(f_i - f_j) + 2\langle Mg_j - mg_i, x_j - x_i\rangle \tag{63}$$

$$- \|g_j - g_i\|^2 - Mm\|x_i - x_j\|^2. \tag{64}$$

There is a whole literature on interpolability of convex functions. In more general cases, this can give rise to things like cyclic monotonicity conditions, which are also related to results such as Brenier's Theorem in optimal transport.

Applying the preceding results, our optimization problem equivalent to a problem we call the *PEP-primal* problem

$$\min_{\substack{\{x_i,g_i,f_i\}_{i=0,1,2,*} \\ Q_{ij}\geq 0, x_0\neq x^* \\ x_1 = x_0 - \alpha\nabla f(x_0) \\ x_2 = x_1 - \beta\nabla f(x_1)}} \frac{\|x_2 - x^*\|}{\|x_0 - x^*\|}. \tag{65}$$

This can be reparametrized as a semi-definite program, which can be solved numerically on a computer. We can also formulate a dual perspective, called the *PEP-dual* problem.

$$\min_{r\geq 0, \{\lambda_i \geq 0\}} \left\{ R : R\|x_0 - x^*\|^2 - \|x_2 - x^*\|^2 = \sum_i \lambda_i Q_{ij} + (\text{square term}) \right\} \tag{66}$$

This is a dual perspective to before: rather than asking what is the worst-case function, we ask what is the best convergence rate. We conclude with some pros and cons:

1. We get exact rates up to numerical precision.
2. We get a semi-definite program, which can be solved.
3. We get solutions for finite n, but not asymptotics.
4. We only get algorithm analysis, not algorithm design.
5. The proofs given can be complicated and are not necessarily interpretable.

One way people use these approaches is to analyze some algorithm for some small n-value, see which inequalities are used, and then try to extrapolate and generalize.

6 Silver step sizes

We ask: can we make gradient descent faster just by changing the step sizes? As before, we work with strongly convex and smooth functions. A priori, it is unclear whether any of the benefits of using different step sizes carry over from the quadratic to the convex case, since the polynomial analysis completely breaks down. We know that:

1. For standard constant step sizes, we get $\mathcal{O}(k)$ rates on quadratic and convex functions.
2. For optimized step sizes, we get $\mathcal{O}(\sqrt{k})$ rates on quadratic functions, but for convex functions rates were not known until recently.
3. With additional dynamics, we get $\mathcal{O}(\sqrt{k})$ rates on quadratic functions by the heavy ball method, and on convex functions by accelerated methods.

Some of the difficulties are that many phenomena are false beyond quadratics: Chebyshev step sizes can diverge, order matters, and things are fundamentally different. It is necessary to track how iterations affect each other, which is only simple in the quadratic case because gradient descent is linear.

The answer, for $n = 2$, is that

$$\alpha_1 = \frac{2}{m+S} \qquad \alpha_2 = \frac{2}{2M+m-S} \tag{67}$$

$$R^* = \frac{S-M}{2m+S-M} \qquad S = \sqrt{M^2+(M-m)^2} \tag{68}$$

which shows $R^* < R_1^2$. Thus, repeating the above scheme periodically gives a constant-factor improvement over the one-step rate. The rate is approximately

$$R^* \approx 1 - \frac{2\rho}{K}\rho = 1+\sqrt{2} \tag{69}$$

where ρ is called the *silver ratio*. Key phenomena are (i) provable advantage of non-constant step sizes, (ii) step size splitting, one alternates between short and long steps, (iii) order matters, (iv), and the shorter and longer step sizes need to be closer to the constant rate than what is used for quadratics.

Why is a constant step size schedule suboptimal? The intuition is that the shorter and longer steps are bad on completely different worst-case functions. We improve because worst cases can't align. Each individual step size is suboptimal alone, but help when used together. We call this *hedging*.

Various papers and numerical investigation showed that larger n values led to larger constant-factor improvement. Following this, extending the two-step solution above using recursion leads to a step size schedule for which one can show an accelerated rate of

$$\mathcal{O}\big(k^{\log_\rho 2}\big). \tag{70}$$

One can define a sequence of ratios via continued fractions

$$n + \frac{1}{n + \frac{1}{n+..}} \tag{71}$$

for which $n = 1$ gives the golden ratio and $n = 2$ gives the silver ratio. The *silver step sizes* are

$$\alpha_t = 1 + \rho^{\nu(t)-1} \tag{72}$$

where ν is the smallest power of 2 in a binary expansion of i. This gives a fractal-like schedule of step sizes, half of which are below the constant rate, and half of which are above. Larger step sizes grow exponentially by a factor of ρ, but become exponentially rarer by a factor of 2.

The analysis involves an idea called *recursive gluing*. Is there is an n-step certificate, then one can copy it twice to get a certificate for $2n$. The idea is to slightly change the valid inequalities in a structured way that gives an improved $2n$-certificate.