Summary of the Classical Methods for Convex and Non-convex Optimization

Qinghua Ding
Computer Science Department
Tsinghua University

Abstract

Recently I've been reading quite a lot papers and books in convex optimization (including operator theory) and non-convex optimization. However, I found it hard to keep track of different methods. Each method seem to have its own constraints and results. Here I will sort ALL IMPORTANT methods out, either in convex optimization and non-convex optimization. These methods are critical to our understanding of optimization methodology and will be helpful for designing better algorithms.

Non-convex Optimization

An area of optimization that is receiving more and more attention is the non-convex optimization, although it is supposed to be already studied a few decades ago and found hard enough (generally NP-hard). Albeit minimizing towards the global optima is hard, new research directions appear as to minimize the first order derivative, or find a local minima.

Specially, we distinguish two kinds of non-convex optimization methods. The first only uses and constrains on first order information, and we'd call it the first-order optimization; while the second, is based on first-order and second-order oracle, targeting at arriving at a local minima, and it's also known in classical optimization theory as nonlinear optimization.

Usually, we consider the following properties of non-convex optimization methods. First comes the convergence, i.e., whether it leads to a stationary point or a local minima. The second is the global convergence rate, i.e., how fast it converges to a global minima. Thirdly, its local convergence rate is also important, especially for some nonlinear optimization methods.

In the following parts, we will introduce some classical methods in either first-order optimization or nonlinear optimization. And we'll give their theoretical performance, with proofs, to make this survey self-contained. Before entering into this problem, we introduce some important preliminary information about non-convex optimization.

Preliminary

Before we handle anything in non-convex optimization, we should always keep in mind the following theorem.

Theorem 0. (*hardness result*) Given 0-1-2 order oracle, finding a global minimizer of some function is generally NP-hard.

In the case where we are only given 0-order oracle, we have the following restrictive theorem, which partially verified Thm0 (proof at [LecConv P7]).

Theorem 1. (*complexity for general 0-order*) Given any function that is L-Lipschitz and we want to find a ϵ-optimal minimizer with only zero order oracle, then the analytical complexity is

$$\mathcal{A} = \left(\left[\frac{L}{2\epsilon} \right] + 2 \right)^n .$$

It's quite noteworthy that this complexity is exponential in dimension n, which is well known as the curse of dimension. Therefore, we cannot expect, in any sense, to find an exact minimizer for a general optimization problem. However, we may still find the approximate minimizer, i.e., (first-order) the point whose gradient is no more than some threshold ϵ; (second-order) or in nonlinear optimization, the point whose gradient is no more than some threshold ϵ and Hessian is almost positive definite.

Since we cannot do any better if we make no more assumption over the function class to be optimized. Thus we consider the continuous function class defined as below. Note that the continuity is almost the most basic property of a well-defined function.

Definition 1. (*Lipschitz continuity*) Let Q be a subset of R^n, then the function $f : Q \to R \in \mathcal{C}_L^{k,p}$ satisfies:

1. $f(\cdot)$ is k times continuous differentiable;
2. $\forall x, y \in Q$, we have $\|f^p(x) - f^p(y)\| \leq L\|x - y\|$.

And we have the following descent lemma, which is critical to convergence analysis.

Lemma 1. (*descent lemma*) Suppose $f : Q \to R^n \in \mathcal{C}_L^{1,1}$. Then we have for $(\forall x \in Q)$ $(\forall y \in Q)$,

$$f(y) \leq f(x) + \langle f'(x) | y - x \rangle + \frac{L}{2} \parallel y - x \parallel^2 .$$

Proof. Consider the following calculus.

$$
\begin{aligned}
f(y)-f(x) &= \int_{u=x}^{y} f'(u)du \\
&= \int_{\tau=0}^{1} f'(x+\tau(y-x))(y-x)d\tau \\
&= \langle f'(x)|y-x\rangle + \int_{\tau=0}^{1} (f'(x+\tau(y-x)) \\
&\quad - f'(x))(y-x)d\tau \\
&\leq \langle f'(x)|y-x\rangle + \int_{\tau=0}^{1} \parallel (f'(x+\tau(y-x)) \\
&\quad - f'(x))(y-x) \parallel d\tau \\
&= \langle f'(x)|y-x\rangle + \int_{\tau=0}^{1} \parallel \tau L \parallel y-x \parallel^2 d\tau \\
&= \langle f'(x)|y-x\rangle + \frac{L}{2} \parallel y-x \parallel^2
\end{aligned}
\tag{1}
$$

For higher order approximation, we also have ($\forall x \in Q$) ($\forall y \in Q$),

$$
\begin{aligned}
f(y) \leq & f(x) + \langle f'(x)|y-x\rangle \\
& + \frac{1}{2}\langle f''(x)(y-x)|(y-x)\rangle + \frac{M}{6} \parallel y-x \parallel^2 .
\end{aligned}
\tag{2}
$$

Two basic idea in classical non-convex optimization is approximation and relaxation. The relaxation is straightforward, that is, to construct a series of array thus it is decreasing and is an upper bound for the minimum of $f(x)$ over Q.

And the approximation says that we approximate the black-box function by some lower-order functions thus we can make use of the information we derived from the oracle, and thus we can update our model.

In the following subsections, we consider three classes of algorithms. The first class only cares about deriving the first-order approximate, we call then first-order methods; the second class caters to the second-order approximate and thus we call then second-order approximate. Then we closed this section by discussion of constrained non-convex optimization methods.

I.1 Gradient Descent

In the first chapter, we consider some basic methods which does not guarantee a local minima convergence (or at least not proved to do so). And we begin by the most well known method of gradient descent. Consider in the nth iteration step, we have collected the first-order information from the oracle. And via first-order approximation, we get

$$\bar{f}_k(x) = f(x_k) + f'(x_k)(x-x_k).$$

Now given this approximation, we hope to find a direction for local update. And its' clear that $-\bar{f}'_k(x) = -f'(x_k)$ gives the antiderivative of the approximation function at x_k. We will use this antiderivative to construct our updating rule.

$$x_{n+1} = x_k - h_k f'(x_k)$$

This updating rule is elegant, and the descent lemma claims that for $h_k = \frac{1}{L}, \forall k$, we have

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L} \parallel f'(x_k) \parallel^2 .$$

Then by telescoping, we have

$$
\begin{aligned}
\sum_{k=0}^{N} \parallel f'(x_k) \parallel^2 \leq & 2L(f(x_0) - f(x_n)) \\
\leq & 2L(f(x_0) - f(x^*))
\end{aligned}
\tag{3}
$$

And it's clear that the minimum gradient in history, $\min_k \parallel f'(x_k) \parallel$ is bounded as

$$\min_k \parallel f'(x_k) \parallel \leq \sqrt{\frac{2L(f(x_0)-f(x^*))}{N+1}}.$$

The local rate of convergence for gradient descent is somewhat better [LecConv P30].

$$\parallel x_0 - x^* \parallel \leq \frac{\bar{r}r_0}{\bar{r}-r_0}\left(1 - \frac{2l}{L+3l}\right)^k$$

This convergence rate is called linear. But in general, we'd say that gradient descent offers sublinear convergence rate.

I.2 Newton's Method

Since approximating the function by first order method is very promising already. And we consider the second order approximation as follows.

$$
\begin{aligned}
\bar{f}_k(x) = & f(x_k) + \langle f'(x_k)|y-x_k\rangle \\
& + \frac{1}{2}\langle f''(x_k)(y-x_k)|(y-x_k)\rangle
\end{aligned}
\tag{4}
$$

Thus by letting the first order derivative of the function above as 0, we get

$$x_{k+1} = x_k - [f''(x_k)]^{-1} f'(x_k).$$

This leads to another updating rule - Newton Method. But it has some drawbacks. It breaks down when $f''(x_k)$ is degenerate. And this Newton process could diverge. And to tackle the divergence problem, practitioners usually adopt a damped Newton method.

$$x_{k+1} = x_k - h_k[f''(x_k)]^{-1} f'(x_k).$$

However, no global convergence results about Newton method is whatsoever made. And the degenerated Hessian can also make the algorithm fragile. Anyway, what makes Newton method stands out as a good method is the fast local convergence, which is quadratic[LecConv P35].

$$\parallel x_{k+1} - x^* \parallel \leq \frac{M \parallel x_k - x^* \parallel^2}{2(l - M \parallel x_k - x^* \parallel)}$$

A very tricky way used in practice is to use gradient descent at the beginning, and then switch to the Newton method as the algorithm progresses.

I.3 Quasi-Newton Method [global convergence and rate?]

The gradient descent and Newton method has made the first few steps towards better gradient methods. And quasi-Newton method, which is based on more delicate approximation of the function, is created. In quasi-Newton (also known as variable metric), we use the following approximation.

$$\begin{aligned}\bar{f}_k(x) = f(x_k) + \langle f'(x_k)|y - x_k\rangle \\ + \frac{1}{2}\langle G_k(y - x_k)|(y - x_k)\rangle\end{aligned} \tag{5}$$

First order constraint gives us

$$x_{k+1} = x_k - G_k^{-1} f'(x_k).$$

Thus the idea behind the quasi-Newton is that we construct G_k progressively thus $G_k \to f''(x^*)$. Or a more robust way is to construct H_k gradually thus $H_k \to f''(x_k)^{-1}$. But how can we generate such a sequence? A necessary condition is that

$$H_{k+1}(f'(x_{k+1}) - f'(x_k)) = x_{k+1} - x_k.$$

It's not hard to get this constraint on H_k by noticing the property of the quadratic interpolation between x_k and x_{k+1}. This quasi-Newton method enjoys a superlinear local convergence rate, i.e., we have

$$\| x_{k+1} - x^* \| \leq const \cdot \| x_k - x^* \| \cdot \| x_{k-n} - x^* \| .$$

Anyway, the global convergence rate should be no better than the gradient descent.

I.4 Conjugate Gradient Method [idea, global convergence?]

Based on the quadratic approximation, we can derive more effective gradient method. For example, the conjugate method is specially designed for quadratic function. The idea behind the conjugate method is that, when the domain is sphere-shaped, the gradient can always attain the optima in just a few steps. However, when the domain is ecliptic-shaped, the gradient descent always reaches the optima while shaking from side to side. Thus, we want to force the optimizer to never go backward, and always try new dimensions.

We defer the proofs of the intention of conjugate gradient method to the appendix, and give the convergence result here. This method enjoys a n-step quadratic convergence, i.e., $\| x_{n+1} - x^* \| \leq const \cdot \| x_0 - x^* \|$. But it's global convergence is no better than the gradient method. It's popular among practitioners in that it has cheap iterations.

However, it's impossible for the conjugate gradient method to produce more than n orthogonal gradients in one iteration. Thus it has to restart every n steps.

II.1 Cubic Regularization Method

Under the assumption that $f(x) \in \mathcal{C}_L^{1,1}$, we cannot hope to get any guarantee about the convergence, i.e., wether it converges to a saddle point, or a local minima. Increasing the order of continuity, we seek for better guarantees of this problem. Thus in the cubic regularization method, we aims to find an approximate local minima and thus we make a stronger assertion about the function class, i.e., we assume that $f(x)\mathcal{C}_L^{2,2}$.

We restate the assumption we made and the target we seek for in finding an approximate local minima.

Assumption 1. The Hessian of function f(x) Lipschitz on $\mathrm{dom} f$, i.e.,

$$\| f''(x) - f''(y) \| \leq M \| x - y \| .$$

And our target is to find some $\bar{x}$ such that

- first-order stationary: $\| f'(\bar{x}) \| \leq \epsilon$,
- second-order guarantee: $-\lambda_n(f''(\bar{x})) \leq \delta$.

Now we formally discuss the idea of cubic regularization. In fact, the cubic regularization extends the gradient mapping's idea to higher orders. Specifically speaking, we have the gradient mapping in this occasion as.

$$\begin{aligned}x_Q(\bar{x};\gamma) = &\arg\min_{x\in Q}[\langle f'(\bar{x})|x - \bar{x}\rangle \\ &+\frac{1}{2}\langle f''(\bar{x})(x - \bar{x})|x - \bar{x}\rangle + \frac{\gamma}{6}\| x - \bar{x} \|^3].\end{aligned} \tag{6}$$

We can use the second-order descent lemma to bound the guaranteed decrease when $\gamma \geq L$ as follows.

$$f(\bar{x}) - f(x_Q(\bar{x};\gamma)) \geq \frac{\gamma}{12}\| \bar{x} - x_Q(\bar{x};\gamma) \|^3$$

Note that the convergence rate of this method is $\mathcal{O}(\epsilon^{-\frac{3}{2}})$ on first-order condition, and $\mathcal{O}(\delta^{-3})$ on second-order condition.

II.2 Trust Region Method [convergence rate?]

We won't deeply touch this group of works, which is enormous rather than large. However, we still discuss some of its ideas and properties. We rethink about Newton method, for which there has been a lot of modification proposed. However, the trust region is based on the idea of measuring the deviation of local approximation, and enlarge or decrease the trusted region for the validity of this local approximation. In formal language, we want to solve in iteration k that

$$\min_{\|s\|\leq\Delta_k}\{w_k(x_k) = f(x_k) + \langle f'(x_k)|s\rangle + \langle f''(x_k)s|s\rangle\}.$$

This subproblem is simple to solve, and we measure the gap between the expected decrease $\delta_e = w_k(x_k) - w_k(x_k + s)$ and the real decrease $\delta_r = f(x_k) - f(x_k + s)$. If this ratio is too low, i.e., $\eta = \frac{\delta_r}{\delta_e} \leq \underline{\eta}$, we suppose that the model is not acceptable, and instead of changing the model, we decrease the diameter of the trusted region Δ_k by γ. Otherwise, we

will accept the model and perform a update step. However, if this ratio is too high, we increase the diameter for making larger steps.

The convergence of trusted region method can be established, both for first-order condition, and for second-order condition. However, although the complexity of this method is claimed to be superlinear, the assumption is somewhat special and we won't make any assertion about the complexity of it. Maybe we'll investigate into this problem later. Anyway, it has been a quite popular nonlinear programming method for decades.

III.1 Penalty Function Method

In the third chapter, we consider constrained minimization, which is $\min f_0(x), s.t. f_i(x) \leq 0, \forall i = 1, ..., m$. Before considering about solutions, we first make it clear that constrained minimization is not generally easier than unconstrained ones. And up to now, I know no method that guarantees the convergence rate even to an approximate optima.

To solve this problem, a basic idea used widely in design of algorithms is that, solving a sequence of unconstrained minimization leading to the exact solution of the constrained minimization, i.e., *sequential unconstrained minimization*. And we first define the penalty function.

Definition 2. A continuous function $\Phi(x)$ is called a penalty function for a closed set Q if

- $\Phi(x) = 0$ for any $x \in Q$,
- $\Phi(x) > 0$ for any $x \notin Q$.

And some commonly used penalty functions include

- Quadratic penalty: $\Phi(x) = \sum_{i=1}^{m} (f_i(x))_+^2$,
- Nonsmooth penalty: $\Phi(x) = \sum_{i=1}^{m} (f_i(x))_+$.

The basic idea of penalty method is clear and simple. We solve a series of penalized unconstrained optimization. And in iteration k, we solve

$$\min f_0(x) + t_k \Phi(x).$$

And it's clear that when $t_k \to +\infty$, we have the penalized minima approximates an optima. And frustratingly, there is no rule for selecting the coefficients or the penalty function, there is no bound on accuracy, there is no rate of convergence established.

III.2 Barrier Function Method

Similarly, we consider another way of constructing such sequence of unconstrained optimization. We first define the barrier function as follows.

Definition 3. For a closed set Q with int$Q \neq \emptyset$, the barrier function $F(x)$ is a continuous function so that $F(x) \to +\infty$ when $x \to$bdry$Q$.

Some frequently used barrier functions include

- Power-function barrier: $F(x) = \sum_{i=1}^{m} \frac{1}{(-f_i(x))^p}, \forall p \geq 1$.
- Logarithmic barrier: $F(x) = -\sum_{i=1}^{m} \ln(-f_i(x))$.
- Exponential barrier: $F(x) = \sum_{i=1}^{m} \exp\left(\frac{1}{-f_i(x)}\right)$.

We introduce the notion of Slater condition here, which guarantees the existence of a barrier function for the non-convex optimization.

Definition 4. (*Slater condition*) For the nonlinear optimization whose feasible set is Q, the Slater condition is satisfied iff. int$Q \neq \emptyset$.

The barrier function method progressively solves

$$\min f_0(x) + \frac{1}{t_k} F(x).$$

And it can be proved that when $t_k \to +\infty$, the barrier optimization converges to the optima. However, the convergence rate still cannot be established.

Convex Optimization

Intrinsically, finding a local minima in the non-convex optimization problems can be quite hard. And in most cases, what we can do is only to find a first-order approximate which is ϵ-close to satisfying the first order condition only. A very straight forward idea of the optimization society is to prove the power of first-order methods by constraining on the function class.

Preliminary

We want function class where first-order condition already leads to minimizers. And to generate our function class, we define the base and the rules. Notice that the linear function is simple and quite satisfactory for the first-order condition and we'd choose it as our base. For the generating rules, we use the basic closedness over linear operations. To sum up, we have the constraints as follows.

- (base) the continuous function class $\mathcal{F}$ contains $f(x) = \alpha + \langle a, x \rangle$,
- (induction) if $f_1, f_2 \in \mathcal{F}$ and $\alpha, \beta \geq 0$, then $\alpha f_1 + \beta f_2 \in \mathcal{F}$,
- (design goal) $f'(x) = 0 \Rightarrow f(x) = \min f(\cdot)$.

The following convex function class is exactly fitting for the conditions above (proof at LecConv P52).

Definition 5. (*convex function class*) The function class $\mathcal{F}^1$ includes all functions $f(\cdot)$ thus $\forall x, y \in R^n$, we have

$$f(y) - f(x) \geq \langle f'(x) | y - x \rangle.$$

It can be verified that $\mathcal{F}^1$ is closed under affine operations, thus $f(Ax + b) \in \mathcal{F}^1$ for $f(x) \in \mathcal{F}^1$. And we have the following conditions equivalent to the definition of convex functions (proof at LecConv P54).

- $\forall x, y \in R^n, \forall \alpha \in [0, 1], f(x_\alpha) \leq \alpha f(x) + (1 - \alpha) f(y)$,
- $\forall x, y \in R^n, \langle f'(x) - f'(y) | x - y \rangle \geq 0$,
- (only for $f \in \mathcal{C}^2$) $\forall x \in R^n, f''(x) \succeq 0$.

To this end, we incorporate Lipschitz continuity to guarantee more tractability. Thus, $\mathcal{F}_L^{1,1}$ denotes the functions that satisfies one of the followings for any $x, y \in R^n$ (proof at LecConv P57).

- (descent) $0 \leq f(y) - f(x) - \langle f'(x) | y - x \rangle \leq \frac{L}{2} \parallel x - y \parallel^2$,

- $f(x) + \langle f'(x)|y-x\rangle + \frac{1}{2L} \parallel f'(x) - f'(y) \parallel^2 \leq f(y)$,
- (nonexpansive) $\frac{1}{L} \parallel f'(x) - f'(y) \parallel^2 \leq \langle f'(x) - f'(y)|x - y\rangle$,
- $\langle f'(x) - f'(y)|x-y\rangle \leq L \parallel x - y \parallel^2$

We'd prove the first inequality. Consider the function $\phi(y) = f(y) - \langle f'(x_0)|y\rangle$. Notice that $\phi \in \mathcal{F}_L^{1,1}$, and it's optimal point is $y^* = x_0$. Therefore, we have

$$\phi(y^*) \leq \phi(y - \frac{1}{L}\phi'(y)) \leq \phi(y) - \frac{1}{2L} \parallel \phi'(y) \parallel^2 .$$

This establishes the first inequality. And a question that arise naturally is that what's the complexity of optimization over this function class? Under the following assumption, we can show that it's also cursed by the dimension.

Assumption 1. An iterative method $\mathcal{M}$ generates a sequence of test points $\{x_k\}$ such that

$$x_k \in x_0 + span\{f'(x_0), ..., f'(x_{k-1})\}, \forall k \leq 1.$$

The following theorem shows the limitation of the iteration methods under Assum.1 on convex optimization.

Theorem 2. (*convex with iterative method*) Under Assum1. to minimize a function $f \in \mathcal{F}_L^{1,1}$ with only first-order oracle, we have the following universal limitation on convergence. For $k \leq \frac{1}{2}(n-1)$,

$$f(x_k) - f^* \geq \frac{3L \parallel x_0 - x^* \parallel^2}{32(k+1)^2}.$$

Thus it would be impossible for the target value to converge faster than sublinear rate if no more than $\frac{1}{2}(n-1)$ iterations has been made, even thought the gradient diminishes linearly. However, when the dimension of the problem is quite large, it would be impossible to even go through $O(n)$ iterations, thus we may need a stronger assumption.

Previously, in deriving local convergence rates under gradient descent and Newton method, a critical assumption made is about the non-degeneracy of the local minima. And now we want to make these methods work efficiently globally, it's straight forward to globalize these non-degeneracy assumption. Thus for any $\bar{x}$, if $f'(\bar{x}) = 0$, then we have that there exists some $\mu > 0$, thus $\forall x \in R^n$,

$$f(x) \geq f(\bar{x}) + \frac{1}{2}\mu \parallel x - \bar{x} \parallel^2 .$$

In fact, we need to add this condition to the convex function class $\mathcal{F}^1$, and we denote it as $\mathcal{S}_\mu^1$, the strongly convex function class. We then have the following equivalent definition of the strongly convex functions.

Definition 6. (*strongly convex function*) A continuous differentiable function $f(x)$ is called strongly convex if $\exists \mu > 0, \forall x, y \in R^n$, we have

$$f(y) \leq f(x) + \langle f'(x)|y-x\rangle + \frac{1}{2}\mu \parallel y - x \parallel^2 .$$

Here μ is called the convexity parameter of function f. We can further verify the following equivalent statements of the strongly convex functions.

$$\langle f'(x) - f'(y)|x-y\rangle \geq \parallel x - y \parallel^2$$

And we have the following results due to strong convexity.

- $f(y) \leq f(x) + \langle f'(x)|y-x\rangle + \frac{1}{2\mu} \parallel y - x \parallel^2$,
- $\langle f'(x) - f'(y)|x-y\rangle \leq \frac{1}{\mu} \parallel f'(x) - f'(y) \parallel^2$.

These properties can be verified in the same way as in the convexity's case. Furthermore, we consider the function that is both strongly convex and Lipschitz continuous, i.e., $f(x) \in \mathcal{S}_{\mu,L}^{1,1}$. We have the following inequality for this function class (proof at LecConv P66).

$$\begin{aligned}\langle f'(x) - f'(y)|x-y\rangle \geq \frac{\mu L}{\mu + L} \parallel x - y \parallel^2 \\ + \frac{1}{\mu + L} \parallel f'(x) - f'(y) \parallel^2\end{aligned} \tag{7}$$

And we denote $Q_f = \frac{L}{\mu} \geq 1$ as the condition number. We then introduce the following approximate result.

Theorem 3. (*strongly convex lower bound*) Given first order oracle, to find a ϵ-approximate optima for a function $f \in \mathcal{F}_{\mu,L}^{1,1}$ has the following convergence lower bound (proof at LecConv P67).

$$f(x_k) - f^* \geq \frac{\mu}{2}\left(\frac{\sqrt{Q_f} - 1}{\sqrt{Q_f} + 1}\right)^{2k} \parallel x_0 - x^* \parallel^2$$

In the following chapters, we will discuss unconstrained convex optimization (chap I) and constrained convex optimization (chap II).

I.1 Gradient Descent

We first consider the gradient descent method's performance over convex and strongly convex functions. We have therefore

$$\Delta_k = f(x_k) - f^* \leq \langle f'(x_k)|x_k - x^*\rangle \leq r_0 \parallel f'(x_k) \parallel .$$

And for the gradient descent, we have

$$\begin{aligned}f(x_{k+1}) &\leq f(x_k) + \langle f'(x_k)|x_{k-1} - x_k\rangle \\ &\quad + \frac{L}{2} \parallel x_{k+1} - x_k \parallel^2 \\ &\leq f(x_k) - w \parallel f'(x_k) \parallel^2\end{aligned} \tag{8}$$

Here $w = h(1 - \frac{L}{2}h)$. Then we have that

$$\Delta_{k+1} \leq \Delta_k - \frac{w}{r_0^2}\Delta_k^2$$

.

And we have $\Delta_{k+1}^{-1} \geq \Delta_k^{-1} + wr_0^{-2}$. And by telescoping, we have $\Delta_{k+1}^{-1} \geq \Delta_0^{-1} + (k+1)wr_0^{-2}$. And we choose $h = \frac{1}{L}$, and using $f(x_0) \leq f^* + \frac{L}{2} \parallel x_0 - x^* \parallel^2$, this lead to the sublinear convergence rate.

$$f(x_k) - f^* \leq \frac{2L \parallel x_0 - x^* \parallel^2}{k+4}$$

And for the strongly convex optimization, we can use the new set of inequalities to achieve a linear convergence rate when $h = \frac{2}{\mu+L}$ (proof at LecConv P70).

$$f(x_k) - f^* \leq \frac{L}{2}\left(\frac{Q_f - 1}{Q_f + 1}\right)^{2k} \parallel x_0 - x^* \parallel^2$$

I.2 Optimal Method [mirror descent? accelerated gradient descent?]

Considering the lower bound and the gradient descent convergence rate, these results, in fact, are an order of magnitude away. The optimal method is born in order to close this gap. The arguement of the optimal method is that, gradient descent is too coarse, and only considered about local gradient informations. Thus the optimal method seeks for a global view of the convex optimization problem, using the global properties of this function class for designing a better algorithm.

The global method comes with the idea that is intrinsically different from relaxation, i.e., the estimate sequence. We denote the estimate sequence for a function $f(x)$ as follows.

Definition 7. (*estimate sequence*) A pair of sequences $\{\phi_k(x)\}$ and $\{\lambda_k\}$ is called an estimate sequence of function $f(x)$ if $\lambda_k \to 0$ and $\forall x \in R^n, \forall k \in N$, we have

$$\phi_k(x) \leq (1 - \lambda_k)f(x) + \lambda_k \phi_0(x).$$

The next statement explains why these objects could be useful.

Lemma 2. If for some sequence $\{x_k\}$, we have $f(x_k) \leq \phi_k^*$, then $f(x_k) - f^* \leq \lambda_k(\phi_0(x^*) - f^*) \to 0$.

In fact, we can view the problem of optimizing $f(x)$ as two main steps - find a sequence of estimate functions ϕ_k, and then find a sequence $\{x_k\}$ that satisfies Lemma 2. To construct a sequence of estimate function, it's not hard to consider the following procedure, which is very similar to the mirror descent style.

Lemma 3. (*construction of estimate sequence*) Assume that $\{y_k\}$ is an arbitrary sequence in R^n, and $\alpha_k \in (0,1), \sum_k \alpha_k = \infty$, then we have the following sequence as an estimate sequence.

- $\lambda_0 = 1, \lambda_{k+1} = (1 - \alpha_k)\lambda_k$,
- $\phi_{k+1}(x) = (1 - \alpha_k)\phi_k(x) + \alpha_k \bar{f}_k(x)$.

Here $\bar{f}_k(x)$ is the second order approximation of $f(x)$ given y_k.

$$\bar{f}_k(x) = f(y) + \langle f'(y_k) | x - y \rangle + \frac{\mu}{2} \parallel x - y \parallel^2$$

It's clear that $\forall k \in N$, we have $f(x) \geq \bar{f}_k(x)$. It's not hard to verify by induction that this is an estimate sequence. And more interestingly, we can consider $\phi_k(x)$ as an approximation function that incorporates the previous lower bounds on $f(x)$ to derive a better lower bound.

However, the construction of x_k is somewhat harder. However, we are free in the choice of both $\{y_k\}, \{\alpha_k\}$ and $\phi_0(x)$. We consider the following case, where we set $\phi_0(x) = \phi_0^* + \frac{\gamma_0}{2} \parallel x - v_0 \parallel^2$, then we have the following claim for the function $\phi_k(x)$.

Lemma 4. When $\phi_0(x) = \phi_0^* + \frac{\gamma_0}{2} \parallel x - v_0 \parallel^2$, the functions $\phi_k(x)$ generated by the estimate sequence follows the canonical form.

$$\phi_k(x) = \phi_k^* + \frac{\gamma_k}{2} \parallel x - v_k \parallel^2$$

This lemma is clear from induction (proof at LecConv P73). In fact, we can derive the update rules as follows.

$$\begin{aligned}
\gamma_{k+1} &= (1 - \alpha_k)\gamma_k + \alpha_k \mu \\
v_{k+1} &= \frac{1}{\gamma_{k+1}}[(1 - \alpha_k)\gamma_k v_k + \alpha_k \mu y_k - \alpha_k f'(y_k)] \\
\phi_{k+1}^* &= (1 - \alpha_k)\phi_k + \alpha_k f(y_k) - \frac{\alpha_k^2}{2\gamma_{k+1}} \parallel f'(y_k) \parallel^2 \\
&+ \frac{\alpha_k(1 - \alpha_k)\gamma_k}{\gamma_{k+1}}(\frac{\mu}{2} \parallel y_k - v_k \parallel^2 + \langle f'(y_k) | y_k - v_k \rangle).
\end{aligned} \tag{9}$$

And accordingly, we can add more constraints on these control variables to derive the sequence $\{x_k\}$. And finally we have one special case of the optimal methods as follows (procedure in LecConv P77).

- choose $y_0 = x_0 \in R^n$.
- for kth iteration, compute

$$\begin{aligned}
x_{k+1} &= y_k - h f'(y_k), \\
y_{k+1} &= x_{k+1} + \eta(x_{k+1} - x_k).
\end{aligned}$$

And $\eta = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$, $h = \frac{1}{L}$ is the parameter. We make some interpretation of this formula. We can treat the term $\eta(x_{k+1} - x_k)$ as a retraction, which keeps x_{k+1} closer to the previous x_k. Thus when the convexity is stronger, the retraction is less important; while when the convexity is weaker, the retraction is more significant. And this is what makes it different from the gradient descent. This method enjoys the optimal convergence rate.

$$\begin{aligned}
f(x_k) - f^* \leq \min &\left\{ \left(1 - \sqrt{\frac{\mu}{L}}\right)^k, \frac{4L}{(2\sqrt{L} + k\sqrt{\mu})^2} \right\} \\
&\times [f(x_0) - f^* + \frac{\mu}{2} \parallel x_0 - x^* \parallel^2]
\end{aligned} \tag{10}$$

II.1 Gradient Mapping for Simple Sets

Before discussing constrained minimization, we first consider which kind of constrained minimization can be efficiently solved. It's not hard to deduce from the definition of convex functions that, if $x, y \in domf$, then $x_\alpha \in domf$. This naturally gives us a definition of the convex set.

We review the problem of $\min_{x\in Q} f(x)$, where Q is a convex set, and $f(x)$ is a convex or a strongly convex function. We have the following theorem giving a sufficient and necessary condition for the optima (proof at LecConv P84).

Theorem 4. Let $f(x) \in \mathcal{F}^1$ and Q be a closed convex set, then x^* is an optima of $f(x)$ over Q iff

$$\langle f'(x^*)|x - x^*\rangle \geq 0.$$

And when the function class is restricted to strongly convex functions, the uniqueness gets guaranteed.

Theorem 5. (*unique minimizer*) Any function $f(x) \in \mathcal{S}^1_\mu$ has an unique minimizer over any closed convex set Q.

However, for bounded convex set, blind gradient could step over the boundary and make the function value meaningless. We step back to think about what we really did in gradient descent, i.e., the descent lemma.

$$f(x) \leq f(\bar{x}) + \langle f'(\bar{x})|x - \bar{x}\rangle + \frac{L}{2} \parallel x - \bar{x} \parallel^2$$

This gives us a guaranteed decrease on target function as $\Delta = f(\bar{x}) - f(x) \geq \langle f'(\bar{x})|\bar{x} - x\rangle + \frac{L}{2} \parallel \bar{x} - x \parallel^2$, we want to maximize the guaranteed decrease, and that's why we choose $x = \bar{x} - \frac{1}{L} f'(\bar{x})$ essentially. Now we turn to the problem of constrained minimization, where we want to solve a similar problem, but constrained on Q.

$$\max_{x\in Q} \left\{ \langle f'(\bar{x})|\bar{x} - x\rangle + \frac{L}{2} \parallel \bar{x} - x \parallel^2 \right\}$$

However, we generalized a little bit to extend the flexibility in the choice of control parameters.

$$\begin{aligned} x_Q(\bar{x};\gamma) =& \arg\min_{x\in Q} f(\bar{x}) + \langle f'(x)|x - \bar{x}\rangle + \frac{\gamma}{2} \parallel x - \bar{x} \parallel^2, \\ g_Q(\bar{x};\gamma) =& \gamma(\bar{x} - x_Q(\bar{x};\gamma)). \end{aligned} \tag{11}$$

We call this $g_Q(\bar{x};\gamma)$ the gradient mapping of f on Q. And it's not hard to see that, when $Q \equiv R^n$, the gradient mapping is in fact $f'(x)$, and $x_Q(\bar{x};\gamma) = \bar{x} - \frac{1}{\gamma} f'(\bar{x})$. But when Q is a closed convex set, the gradient mapping could behave more carefully so that the boundary won't get crossed. And we can still view this $g_Q(\bar{x};\gamma)$ as a constrained gradient on Q, which only differs from $f'(x)$ when it's near the boundary of Q.

Similarly, we can establish the descent analysis bu the following lemma.

Lemma 5. (*gradient mapping*) Let $f(x) \in \mathcal{S}^{1,1}_{\mu,L}$, $\gamma \geq L$ and $\bar{x} \in R^n$. Then for any $x \in Q$, we have

$$\begin{aligned} f(x) \geq& f(x_Q(\bar{x};\gamma)) + \langle g_Q(\bar{x};\gamma), x - \bar{x}\rangle \\ &+ \frac{1}{2\gamma} \parallel g_Q(\bar{x};\gamma) \parallel^2 + \frac{\mu}{2} \parallel x - \bar{x} \parallel^2 . \end{aligned} \tag{12}$$

This can be easily proved using Thm4. and definitions of gradient mapping. We are interested in two special cases where we let $x = \bar{x}$ and $x = x^*$ in the inequality above and produce

$$f(x_Q(\bar{x};\gamma)) \leq f(\bar{x}) - \frac{1}{2\gamma} \parallel g_Q(\bar{x};\gamma) \parallel^2,$$

$$\langle g_Q(\bar{x};\gamma)|\bar{x} - x^*\rangle \geq \frac{1}{2\gamma} \parallel g_Q(\bar{x};\gamma) \parallel^2 + \frac{\mu}{2} \parallel x^* - \bar{x} \parallel^2 .$$

These two inequalities are critical to the convergence of the gradient mapping method. And it's not hard to prove that the analysis complexity results aligned exactly with the unconstrained case, which is linear rate under strong convexity and sublinear under general convexity. But we will also note the assumption on which this conclusion is based.

Assumption 2. (*quadratic optimization*) Solving the quadratic minimization problem over convex set is easy.

This can be easily verified in the unconstrained setting, where both Newton method and conjugate gradient can solve this problem perfectly. And in the constrained setting, we still need to verify this assumption carefully. Later we will see that simple simplex method will satisfy our needs if the constraints are linear, or interior point method when the constraints are quadratic. However, no general assertion can be made to this assumption, in the cases where the gradient mapping is hard to compute, we still need to resort to other methods.

II.2 Functional Constrained Optimization [intuition?]

However, we want to consider more than just convex optimization over simple sets. Generally speaking, we want to consider the following problem.

$$\begin{aligned} &\min f_0(x), \\ &\text{s.t. } f_i(x) \leq 0, i = 1...m. \end{aligned} \tag{13}$$

Since $f_i(x) \leq 0, i = 1...m.$ is equivalent to $g(x) \leq 0$, where $g(x) = \max\{f_i(x), i = 1...m\}$. And we denote t^* as the an optimal value of this problem, then

Here the functions are all convex and smooth, i.e., $f_i(x) \in \mathcal{S}^{1,1}_{\mu,L}$. The key idea for solving this problem lies in transforming this programming into a minimax problem.

$$f^*(t) = \min_{x\in R} \max\{f_0(x) - t, f_i(x), i = 1...m\}$$

And we have the following lemma (proof at LecConv P101).

Lemma 6. $f^*(t) \leq 0$ for $t \geq t^*$; $f^*(t) > 0$ for $t < t^*$.

Thus using this lemma, we essentially transformed the original problem into finding the smallest root of $f^*(t)$, and the subproblem as a minimax problem.

Solving the minimax problem To solve the general minimax problem, $\min_x[f(x) = \max\{f_i(x), i = 1...m\}]$, we consider a linearization technique, i.e., we let $f(\bar{x};x) = \max_{1\leq i\leq m}[f_i(\bar{x}) + \langle f_i'(\bar{x})|x - \bar{x}\rangle]$. And we then have the following lemma.

Lemma 7. Under strongly convexity, we have that a point $x^*$is a solution to $\min f(x)$ iff $\min f(x^*;x) = f(x^*;x^*) = f(x^*)$.

Thus we can now illustrate the main process of handling the minimax problem. First, we linearize $f(x)$ at x_k to get

$f(x_k; x)$, then we calculate the derivative of the function $f(x_k; x)$, and update towards x_{k+1} using this gradient. However, there are still a few problems, for example, when the function $f(x_k; x)$ becomes non-smooth, it becomes impossible to calculate the derivative. Thus instead of using the derivative, we use a gradient mapping as follows.

$$x_f(\bar{x}; \gamma) = \arg\min_x [f(\bar{x}; x) + \frac{\gamma}{2} \parallel x - \bar{x} \parallel^2]$$

$$g_f(\bar{x}; \gamma) = \gamma(\bar{x} - x_f(\bar{x}; \gamma))$$

And also note that, this is based on the assumption that minimizing a max-type linear function adding a quadratic term is easy.

Calculating the root Then the second problem is to calculate a root, given that the function value $f^*(t)$ can be easily evaluated. In fact, we can prove that $f^*(t)$ is non-increasing and is 1-Lipschitz continuous. Moreover, we can also prove that its derivative also increases (proof at LecConv P101).

Lemma 7. $f^*(t)$ is non-increasing and 1-Lipschitz continuous. And $\forall t_0 < t_1 < t_2$, we have

$$f^*[t_0, t_1] \leq f^*[t_1, t_2]$$

This helps us to design efficient schemes for calculating the root. Anyway, the two-level optimization is still quite complex. We'd resort the overall analysis later (find it at LecConv P103).

Non-smooth Convex Optimization

In fact, the function can be quite non-smooth, for example, the max-type convex function can be non-smooth at the intersections of sub functions. In other optimization problems, we can also encounter target functions that are non-smooth, but still has convex properties. Thus we wish to extend the convexity property to the non-smooth cases. However, to handle non-smooth function, we not only have to extend the notion of convexity, but also the notion of differentiability, and the notion of mapping.

Preliminary

We discuss about general convexity, projection and subgradient sequentially. They are important tools for non-smooth optimization.

PreI. General convexity We give the definition of general convexity as follows.

Definition 8. (*general convexity*) A function $f(x)$ is general convex if $\forall x, y \in \text{dom} f$, we have $x_\alpha \in \text{dom} f$ and $f(x_\alpha) \leq \alpha f(x) + (1 - \alpha) f(y)$.

We have some equivalent definitions for general convexity, for example, the epigraph is a convex set. We have the following very important theorems concerning the continuity and differentiability property of the convex functions.

Theorem 6. (*local boundedness*) For f be convex and $x \in \text{int}(\text{dom} f)$, f is locally upper bounded at x_0.

Theorem 7. (*local continuity*) For f be convex and $x_0 \in \text{dom} f$, f is locally Lipschitz continuous at x_0.

Theorem 8. (*local differentiability*) Convex function f is differentiable in any direction at any interior point of its domain.

These are classical results and theirs proofs can be found at the [LecConv].

PreII. Projective Mapping Let Q be a closed convex set, then the projection (or projective mapping) of $x_0 \in R^n$ on Q is

$$\pi_Q(x_0) = \arg\min\{\parallel x - x_0 \parallel : x \in Q\}.$$

And we have the following classical criterion for the projection.

Theorem 9. (*projection*) The projection of x on a closed convex set Q is unique, and for any $x \in Q$, we have

$$\langle \pi_Q(x_0) - x_0 | x - \pi_Q(x_0) \rangle \geq 0.$$

Projection is a helpful tool when we want to get the point that exceeds Q due to gradient descent back to Q.

PreIII. Subgradient We then define the subgradient, which is an extension from its smooth counterpart as derivatives. The subgradient essentially indicates the direction for function decrease. We first consider the definition of subgradient, and then we show how the subgradient can be computed efficiently.

Definition 9. (*subgradient*) The subgradient g of function f at a point $x_0 \in \text{dom} f$ satisfies that for any $x \in \text{dom} f$,

$$f(x) \geq f(x_0) + \langle g | x - x_0 \rangle.$$

Then a natural question is that wether every convex function has nonempty subgradient at any point of its domain. The answer is affirmative (proof at P127).

Lemma 9. Closed convex function f has nonempty subgradient at any point $x_0 \in \text{dom} f$.

And the following theorem establishes the connection between the subgradient and the directional derivative.

Theorem 10. Let f be a closed convex function, $\forall x_0 \in \text{int}(\text{dom} f)$ and $p \in R^n$ we have

$$f'(x_0; p) = \max\{\langle g, p \rangle | g \in \partial f(x_0)\}$$

This says that the directional derivative is in fact the largest projection attainable of some subgradient at this direction. Subgradient can be computed easily for most kinds of functions and their combinations. We give some examples below.

- (linear combination) let $f = \alpha_1 f_1 + \alpha_2 f_2$ be convex and closed, then $\partial f = \alpha_1 \partial f_1 + \alpha_2 \partial f_2$,
- (affine transformation) let $f(x) = g(Ax + b)$ be convex and closed, then $\partial f(x) = A^T \partial g(Ax + b)$,
- (max-type function) let function $f(x) = \max_{1 \leq i \leq m} f_i(x)$, then $\forall x \in \text{int (dom } f)$, we have $\partial f(x) = \text{Conv}\{\partial f_i(x) | i \in I(x)\}$, where $I(x) = \{i | f_i(x) = f(x)\}$.

PreIV. Misc. Also, we give the Kuhn-Tucker optimality condition as follows (proof at LecConv P134).

Theorem 11. (*Kuhn-Tucker*) Let $f_i(x)$ be differentiable convex functions, and the Slater condition is satisfied, then x^* is a solution to functional constrained minimization problem iff $\exists \lambda_i \geq 0, i = 1...m$, such that

$$f_0'(x^*) + \sum_{i \in I^*} \lambda_i f_i'(x^*) = 0,$$

here $I^* = \{i \in [1, m] : f_i(x^*) = 0\}$.

Now we are ready to consider the general non-smooth optimization schemes.

I.1 Subgradient Method

The idea of subgradient method is very clear - using subgradients rather than gradients in gradient descent. Before discussing about the details, we first give a complexity lower bound for the iterative subgradient methods on this problem, which is $\mathcal{O}(\frac{1}{\epsilon^2})$.

And we'd still consider the subgradient's property as the following inequality, which means that the target function decreases in the $-g(x)$ direction.

$$\langle x - x^* | g(x) \rangle \geq 0$$

And this is essentially why we can use the subgradient method. Now we are going to proceed to the subgradient method as follows.

$$x_{k+1} = x_k - h_k \frac{g(x_k)}{\parallel g(x_k) \parallel}$$

Note that the subgradient is normalized to make the method more robust and avoid possible oscillation. We denote $v_i = \frac{g(x_k)}{\|g(x_k)\|^2}$ as this normalized subgradient direction. And we can bound $r_{k+1}^2 = \parallel x_{k+1} - x^* \parallel^2$ as follows.

$$\begin{aligned} r_{k+1}^2 &= \parallel x_{k+1} - x^* \parallel^2 \\ &= \parallel x_k - x^* - h_k v_k \parallel^2 \\ &= r_k^2 + h_k^2 - 2h_k v_k \\ &= r_0^2 + \sum_{i=0}^{k} h_i^2 - \sum_{i=0}^{k} 2h_i v_i \end{aligned} \tag{14}$$

Anyway, it would be hard for us to move any further as in gradient descent. We have to stop and consider how we can bound the $\langle r_i | v_i \rangle$ term. This term is essentially $\parallel g(x_k) \parallel^{-1} \langle x_k - x^* | g(x_k) \rangle$, and this term is strictly larger than zero. But we still need better bounds on this term to determine how much decrease the subgradient method has made.

To do this, we have to introduce a new conception as localization, since the method used in smooth optimization like relaxation and approximation would hardly help here. The basic idea of localization is as follows. We interpret the information of $\langle x - x^* | g(x) \rangle \geq 0$ in another way. In fact, it not only claims the direction $-g(x)$ to be a decreasing direction, but it also constrained on the x^*, so that it has to be on the right halfspace. And the intuition is that when we have enough $x_i, g(x_i)$ pairs, we can effectively bound the space that x^* could reside in, and therefore we can make a good localization of this x^*.

If the closeness to the optimal solution is bounded, we can then bound the closeness of target function by the local Lipschitz property of the convex function. The key ingredient is that for $\bar{x}$ that satisfies $\langle g(x) | x - \bar{x} \rangle \geq 0$, we have $f(x) - f(\bar{x}) \leq w_f(\bar{x}; v_f(\bar{x}; x))$. Here v_f measures the distance from $\bar{x}$ to the separation plane. And w_f measures the largest $f(x') - f(\bar{x})$ within $B(\bar{x}; v_f(\bar{x}; x))$. The intuition of this property is that the target function faces no loss in the direction perpendicular to $g(x)$, and thus a perpendicular ball from $\bar{x}$ to the plane would bound the target difference effectively.

Combining the above arguements, we conclude that for a set of (x_i, f_i, g_i), let $f_k^* = \min_i f_i$, $v_k^* = \min_i v_f(x^*; x_i)$, then the target difference is well bounded by the closeness of x^* and the supporting planes.

$$f_k^* - f^* \leq w_f(x^*; v_k^*) \leq L v_k^*$$

Turning back to the subgradient problem, we have that

$$r_{k+1}^2 \leq r_0^2 + \sum_{i=0}^{k} h_i^2 - \sum_{i=0}^{k} 2h_i v_k^*.$$

And this leads to $v_k^* \leq \frac{R^2 + \sum_{i=0}^{k} h_i^2}{\sum_{i=0}^{k} 2h_i}$, and we finally have

$$f_k^* - f^* \leq L \frac{R^2 + \sum_{i=0}^{k} h_i^2}{\sum_{i=0}^{k} 2h_i}.$$

By choosing $h_i = \frac{r}{\sqrt{i+1}}$, we can achieve convergence rate of $\mathcal{O}(\epsilon^{-2})$, which meets the theoretical bound.

II.2 Cutting Plane Schemes

All cutting plane schemes, including method of center-of-gravity and ellipsoid method require the dimension of x to be limited. And the theoretical complexity lower bound is $\mathcal{O}(n \ln \frac{1}{\epsilon})$. In this case, the localization set $S_0(X) = Q$, $S_{k+1}(X) = \{x \in S_k(X) | \langle g(x_k) | x_k - x \rangle \geq 0\}$ are measurable. And in fact, we have

$$v_k^* \leq D \left(\frac{\text{vol}_n S_k(X)}{\text{vol}_n Q} \right)^{\frac{1}{n}}.$$

The proof can be found at [LecConv P150]. And this is the motivation of acceleration. We can control the halfspace generation process and achieve linear convergence of $\{vol_n S_k(X)\}$, which translates into linear convergence of $\{v_k^*\}$ directly. Thus we have the following process, known as center-of-gravity method.

- Set $S_0 = Q$;
- For kth iteration:
- choose $x_k = cg(S_k)$;
- set $S_{k+1} = \{x \in S_k | \langle g(x_k) | x_k - x \rangle \geq 0\}$.

The convergence rate is guaranteed by the property of center-of-gravity.

$$\frac{\text{vol}_n S_{k+1}}{\text{vol}_n S_k} \leq 1 - \frac{1}{e}$$

This leads to the linear rate of convergence

$$f_k^* - f^* \leq MD(1 - \frac{1}{e})^{-\frac{k}{n}}.$$

However, computing the center of gravity is quite difficult, and making this method absolutely impractical. We consider another bounding technique, i.e., we construct a sequence of ellipsoids rather than a series of simplex. Consider the following ellipsoid.

$$E(H, \bar{x}) = \{x \in R^n | \langle H^{-1}(x - \bar{x}) | x - \bar{x} \rangle \leq 1\}$$

Then with the cutting plane, we have the half ellipsoid as

$$E_+ = \{x \in E(H, \bar{x}) | \langle g | \bar{x} - x \rangle\}.$$

And moreover, we can construct another ellipsoid containing this one, i.e, we let

$$\bar{x}_+ = \bar{x} - \frac{1}{n+1} \cdot \frac{Hg}{\langle Hg | g \rangle^{\frac{1}{2}}};$$

$$H_+ = \frac{n^2}{n^2 - 1} \left(H - \frac{2}{n+1} \cdot \frac{Hgg^T H}{\langle Hg | g \rangle} \right).$$

Then $E_+ \subset E(H_+, \bar{x}_+)$, and this produces linear convergence rate.

$$\text{vol}_n E(H_+, \bar{x}_+) \leq \left(1 - \frac{1}{(n+1)^2}\right)^{\frac{n}{2}} \text{vol}_n E(H, \bar{x})$$

The analytical complexity of the ellipsoid method is $\mathcal{O}(n^2 \ln \frac{1}{\epsilon})$.

II.3 Kelly Method

The Kelly method tries to build a model for the function, which is coarse approximation of the function, with historic information integrated into this estimation.

$$\bar{f}_k(X; x) = \max_i [f(x_i) + \langle g(x_i) | x_i - x \rangle]$$

It's clear that this function is a global lower bound for $f(x)$. The idea of Kelly method is simple. At kth iteration, it chooses x_{k+1} that minimizes this $\bar{f}_k(X; x)$. Anyway, it is quite misleading to do so, in fact. And it is quite unstable. It can be proved that it does not perform any better than ellipsoid method.

II.4 Mirror Descent

The idea of mirror descent is somewhat similar to the optimal method, which encourages a global view for a better algorithm. We consider the supporting plane at the iteration k as

$$\bar{f}_k(x) = f(x_k) + \langle \partial f(x_k) | x - x_k \rangle.$$

And this serves as a lower bound for the original problem, i.e., $f(x) \geq \bar{f}_k(x), \forall x, k$. Thus we have the following intuition (just for analysis). We combine the history together to get a global lower bound, $g_k(x) = \frac{1}{k} \sum_{i=0}^k \bar{f}_k(x) \leq f(x)$. And in each step, we select the minimizer of this function to be the next point. However, to keep the uniqueness of this minimizer, we choose a regularizer, usually quadratic function, and add it to the approximation, $\bar{g}_k(x) = g_k(x) + \phi(x) = \frac{1}{k} \sum_{i=0}^k \bar{f}_k(x) \leq f(x) + \frac{\gamma}{2} \parallel x - x_k \parallel^2$. This regularization can also be considered as a retraction.

To this end, we have discussed the crude idea for the mirror descent. However, full illustration of the mirror descent will need the notion of distance generating function, Bregman divergence, etc., which are very important tools for online convex optimization. Anyway, we still list out the convergence results here. The convergence rate of the mirror descent is $\frac{1}{\sqrt{T}}$, which matches that of the subgradient method.

II.5 Level Methods

The level method (LecConv P155) also achieves the convergence rate as the subgradient method. We do not discuss it for now.

Open Problems

Here we keep track of some critical open problems in optimization theory.

Non-convex Optimization

- What is the lower bound of analytical complexity of nonlinear / first-order optimization?
- How should we choose the best method for constrained nonlinear optimization?

Non-smooth Optimization

- Can we use momentum in non-smooth optimization?
- Can we implement optimal methods $\mathcal{O}(n \ln \frac{1}{\epsilon})$?