# Techniques in Optimization and Sampling
(book in progress)

Yin Tat Lee[1]

University of Washington
& Microsoft Research

Santosh Vempala[2]

Georgia Tech
& Funtime Research

April 6, 2019

[1]yintat@uw.edu. This work is supported in part by CCF-1740551, CCF-1749609, and DMS-1839116.

[2]vempala@gatech.edu. This work is supported in part by CCF-1563838, E2CDA-1640081, CCF-1717349 and DMS-1839323.

# Contents

[1]Anything marked with * means it is probably too mathematical and could be skipped.

# Preliminaries

We use $B(x,r)$ to denote a $\ell_2$ ball centered at $x$ with radius $r$. We use $\text{conv}(X)$ to denote the convex hull of $X$, namely $\text{conv}(X) = \{\sum \alpha_i x_i : \ \alpha_i \geq 0, \sum \alpha_i = 1, x_i \in X\}$. We use $\langle x, y\rangle$ to denote the $\ell_2$ inner product of $x$ and $y$.

## Functions

**Definition.** A function $f : V \to W$ is $L$-Lipschitz if $\|f(x) - f(y)\|_W \leq L\|x - y\|_V$ where the norms $\|\cdot\|_V$ and $\|\cdot\|_W$ are $\ell_2$ norm if unspecified.

**Definition.** A function $f \in \mathcal{C}^k(\mathbb{R}^n)$ if $f$ is a $k$-differentiable function and its $k^{\text{th}}$ derivative is continuous.

## Linear Algebra

**Definition.** A symmetric matrix $A$ is positive semi-definite (PSD) if $x^\top Ax \geq 0$ for all $x \in \mathbb{R}^n$. We write $A \succeq B$ if $A - B$ is PSD.

**Definition.** For any matrix $A$, we define $\text{tr}A = \sum A_{ii}$, $\|A\|_F^2 = \text{tr}(A^\top A)$, $\|A\|_{\text{op}} = \max_{\|x\|_2 \leq 1} \|Ax\|_2$.

For symmetric $A$, we have $\text{tr}A = \sum_i \lambda_i$, $\|A\|_F^2 = \sum_i \lambda_i^2$ and $\|A\|_{\text{op}} = \max_i |\lambda_i|$ where $\lambda_i$ are the eigenvalues of $A$.

## Calculus

**Theorem 0.0.1** (Taylor's Remainder Theorem)**.** *For any $g \in \mathcal{C}^{k+1}(\mathbb{R})$, and any $x$ and $y$, there is a $\zeta \in [x, y]$ such that*

$$g(y) = \sum_{j=0}^{k} g^{(j)}(x)\frac{(y-x)^j}{j!} + g^{(k+1)}(\zeta)\frac{(y-x)^{k+1}}{(k+1)!}.$$

## Probability

**Definition 0.0.2.** The KL-divergence of a density $\rho$ with respect to another density $\nu$ is

$$D_{\text{KL}}(\rho\|v) = \int \rho(x)\log\frac{\rho(x)}{\nu(x)}dx.$$

**Lemma 0.0.3** (Itô's lemma*)**.** *For any process $x_t \in \mathbb{R}^n$ satisfying $dx_t = \mu(x_t)dt + \sigma(x_t)dW_t$ where $\mu(x_t) \in \mathbb{R}^n$ and $\sigma(x_t) \in \mathbb{R}^{n\times m}$, we have that*

$$df(x_t) = \nabla f(x_t)^\top \mu(x_t)dt + \nabla f(x_t)^\top \sigma(x_t)dW_t + \frac{1}{2}\text{tr}(\sigma(x_t)^\top \nabla^2 f(x_t)\sigma(x_t))dt.$$

# Chapter 1

# Introduction

Convexity and its natural extensions are a current frontier of tractable, i.e., polynomial-time, computation. The assumption of convexity induces structure in instances that makes them amenable to efficient algorithms. For example, the local minimum of a convex function is a global minimum. Convexity is maintained by natural operations such as intersection (for sets) and addition (for functions). Perhaps less obvious, but also crucial, is that convex sets can be approximated by ellipsoids in various ways.

In this course, we will study two topics involving convexity, namely optimization and sampling. Given a convex function $f$, (1) how fast can we minimize $f(x)$? (2) how fast can we sample a point according to the distribution $e^{-f(x)}$? These problems are quite closely connected, e.g., sampling from such distributions can be used to find near-optimal points. We will learn several techniques that lead us to some polynomial-time algorithms for both problems and linear-time algorithms for the case $f$ is close to a quadratic function.

Although convex optimization has been studied since the 19th century[1] with many tight results emerging, there are still many basic open problems. Here is an example:

**Open Problem.** Given an $n \times n$ random 0/1 matrix $A$ with $O(n)$ nonzero entries, can we solve $Ax = b$ in less than $n^2$ time?

Computing the volume is an ancient problem, the early Egyptians and Greeks developed formulas for specific shapes of interest. Unlike convex optimization, even computing the volume of a convex body is intractable, as we will see later. Nevertheless, there are efficient randomized algorithms that can estimate the volume to arbitrary accuracy in time polynomial in the dimension and the desired accuracy. This extends to efficient algorithms for integrating logconcave functions.

## 1.1 Definitions

**Definition 1.1.1.** A set $K$ in $\mathbb{R}^n$ is convex if for every pair of points $x, y \in K$, we have $[x, y] \subseteq K$.

**Definition 1.1.2.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is

1. *Convex* if for any $\lambda \in [0,1]$, we have $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$.
2. *Concave* if for any $\lambda \in [0,1]$, we have $f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda) f(y)$.
3. *Logconcave* if $\log f$ is concave, i.e., $f$ is nonnegative and for any $\lambda \in [0,1]$, we have $f(\lambda x + (1-\lambda)y) \geq f(x)^{\lambda} f(y)^{1-\lambda}$.
4. *Quasiconvex* if the levels sets of $f$, $\{x : f(x) \leq t\}$ are convex sets.

An optimization problem $\min_{x\in K} f(x)$ is convex if $K$ and $f$ are convex.

[1] Augustin-Louis Cauchy introduced gradient descent in 1847.

## 1.2 Why is Convexity Useful? Separation!

Any point not in a convex set can be separated by a hyperplane from the set. Such a separating hyperplane allows us to do binary search to find a point in a convex set. This is the basis of all polynomial-time algorithms for optimizing general convex functions. We will explore these binary search algorithms in Chapter 2.

**Theorem 1.2.1.** *Let $K$ be a closed convex set in $\mathbb{R}^n$ and $y \notin K$. There is a non-zero $\theta \in \mathbb{R}^n$ such that*

$$\langle \theta, y \rangle > \max_{x \in K} \langle \theta, x \rangle .$$

*Proof.* Let $x^*$ be a point in $K$ closest to $y$, namely $x^* \in \arg\min_{x \in K} \|x-y\|^2$ (such a minimizer always exists for closed convex sets; this is sometimes calle Hilbert's projection theorem). Using convexity of $K$, for any $x \in K$ and any $0 \le t \le 1$, we have that

$$\|(1-t)x^* + tx - y\|^2 \ge \|x^* - y\|^2 .$$

Expand the LHS, we have

$$\begin{aligned} \|(1-t)x^* + tx - y\|^2 &= \|x^* - y + t(x - x^*)\|^2 \\ &= \|x^* - y\|^2 + 2t \langle x^* - y, x - x^* \rangle + O(t^2). \end{aligned}$$

Taking $t \to 0^+$, we have that

$$\langle x^* - y, x - x^* \rangle \ge 0 \text{ for all } x \in K. \tag{1.2.1}$$

Taking $\theta = y - x^*$, we have that

$$\langle \theta, y \rangle > \langle \theta, x^* \rangle \ge \langle \theta, x \rangle \text{ for all } x \in K.$$

where we used that $\langle \theta, y \rangle - \langle \theta, x^* \rangle = \|\theta\|^2 > 0$ and $\langle \theta, x^* \rangle - \langle \theta, x \rangle = \langle y - x^*, x^* - x \rangle \ge 0$ due to (1.2.1). □

This theorem shows that a polytope (a finite intersection of halfspaces) is essentially as general as a convex set.

**Corollary 1.2.2.** *Any closed convex set $K$ can be written as the intersection of halfspaces as follows*

$$K = \bigcap_{\theta \in \mathbb{R}^n} \left\{ x : \ \langle \theta, x \rangle \le \max_{y \in K} \langle \theta, y \rangle \right\}.$$

*In other words, any convex set is a limit of a sequence of polyhedra.*

Similar to convex sets, we have a separation theorem similar to Theorem 1.2.1 for convex functions. This shows that one can use binary search to find minimum of convex functions. (See Chapter 2.)

**Theorem 1.2.3.** *Let $f \in \mathcal{C}^1(\mathbb{R}^n)$ be convex. Then, for any $x, y \in \mathbb{R}^n$, we have*

$$f(y) \ge f(x) + \nabla f(x)^\top (y - x).$$

*Proof.* Fix any $x, y \in \mathbb{R}^n$. Let $g(t) = f((1-t)x + ty)$. Since $f$ is convex, so is $g$. Then, we have

$$g(t) \le (1-t) g(0) + t g(1)$$

which implies that

$$g(1) \ge g(0) + \frac{g(t) - g(0)}{t}.$$

Taking $t \to 0^+$, we have that $g(1) \ge g(0) + g'(0)$. In other words, using the chain rule for derivatives, we have

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

□

This theorem shows that $\nabla f(x) = 0$ (local minimum) implies $x$ is a global minimum.

**Theorem 1.2.4** (Optimality condition for unconstrained problems)**.** *Let $f \in \mathcal{C}^1(\mathbb{R}^n)$ be convex. Then, $x$ is the minimizer of* $\min_{x \in \mathbb{R}^n} f(x)$ *if and only if* $\nabla f(x) = 0$.

*Proof.* If $\nabla f(x) \neq 0$, then $f(x - \varepsilon \nabla f(x)) = f(x) - (\varepsilon \pm o_\varepsilon(1)) \|\nabla f(x)\|^2 < f(x)$ for small enough $\varepsilon$. Hence, such a point cannot be the minimizer.

On the other hand, if $\nabla f(x) = 0$, Theorem 1.2.3 shows that

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) = f(x) \text{ for all } y.$$

$\square$

We note that the proof above is in fact a constructive proof. If $x$ is not a minimum, it suggests a point that has better function value. This will be the discussion of an upcoming section.

## 1.3 Examples of Convex Sets and Functions

There are many important convex sets and here we only list some that appear in this course. One of the most important classes of convex functions is convex sets.

**Definition 1.3.1.** Given a convex set $K$, we define

$$\delta_K(x) = \begin{cases} 0 & \text{if } x \in K \\ +\infty & \text{otherwise} \end{cases}.$$

If we let $\text{dom} f \overset{\text{def}}{=} \{x \in \mathbb{R}^n : \ f(x) < +\infty\}$, then the definition of convex function shows that $\text{dom} f$ is a convex set. By looking at the set of points above the graph of the function, we obtain a convex set called an epigraph.

**Definition 1.3.2.** The epigraph of $f : \mathbb{R}^n \to \mathbb{R}$ is $\text{epi} f \overset{\text{def}}{=} \{(x,t) \subset \mathbb{R}^n \times \mathbb{R} : \ t \geq f(x)\}$.

**Exercise 1.3.3.** A function $f$ is convex if and only if $\text{epi} f$ is a convex set.

This characterization shows that $\min_x f(x)$ is same as $\min_{(t,x) \in \text{epi} f} t$. Therefore, convex optimization is same as finding an extreme point of a convex set. Another important feature of convex set is the following:

**Exercise 1.3.4.** The sublevel set $\{x \in \mathbb{R}^n : \ f(x) \leq t\}$ is convex.

In particular, this shows that the set of minimizers is connected. Therefore, any local minimum is a global minimum. We note that the converse is not true. A function is *quasiconvex* if every sublevel set is convex.

Finally, we note that many operations preserve convexity. Here is an example.

**Exercise 1.3.5.** Given a matrix $A$, a vector $b$, positive scalars $t_1, t_2 \geq 0$, convex functions $f_1$ and $f_2$. Then $g(x) = t_1 f_1(Ax + b) + t_2 f_2(x)$ is convex.

Here are some convex sets and functions.

**Example.** Convex sets: polytope $\{Ax \leq b\}$, ellipsoid $\{x^\top Ax \leq 1\}$, positive semidefinite cone $\{A \in \mathbb{R}^{n \times n} : \ A \succeq 0\}$, norm ball $\{x : \ \|x\|_p \leq 1\}$ for all $p \geq 1$.

**Example.** Convex functions: $x$, $\max(x, 0)$, $e^x$, $x^a$ for $a \geq 1$, $-\log(x)$, $x \log x$, $\|x\|_p$ for $p \geq 1$, $(x,y) \to \frac{x^2}{y}$, $A \to -\log\det A$ over PSD matrices $A$, $(x, Y) \to x^\top Y^{-1} x$, $\log \sum_i e^{x_i}$, $(\prod_i x_i)^{\frac{1}{n}}$.

## 1.4 Examples of Logconcave Functions and Problems

**Example 1.4.1.** The indicator function of a convex set $1_K(x) = \begin{cases} 1 & \text{if } x \in K \\ 0 & \text{otherwise} \end{cases}$ is logconcave.

**Lemma 1.4.2** (Dinghas; Prékopa; Leindler)**.** *The product, minimum and convolution of two logconcave functions is also logconcave; in particular, any linear transformation or any marginal of a logconcave density is logconcave; the distribution function of any logconcave density is logconcave.*

We next describe the basic theorem underlying the above properties.

**Theorem 1.4.3** (Prékopa-Leindler)**.** *Fix $\lambda \in [0,1]$. Let $f, g, h : \mathbb{R}^n \to \mathbb{R}_+$ be functions satisfying $h(\lambda x + (1-\lambda)y) \geq f(x)^\lambda g(x)^{1-\lambda}$ for all $x \in \mathbb{R}^n$. Then,*

$$\int_{\mathbb{R}^n} h \geq \left(\int_{\mathbb{R}^n} f\right)^\lambda \left(\int_{\mathbb{R}^n} g\right)^{1-\lambda}.$$

An equivalent version of the lemma for sets in $\mathbb{R}^n$ is often useful.

**Theorem 1.4.4** (Brunn-Minkowski)**.** *For any $\lambda \in [0,1]$ and measurable sets $A, B \subset \mathbb{R}^n$, we have*

$$\mathrm{vol}(\lambda A + (1-\lambda)B)^{1/n} \geq \lambda \mathrm{vol}(A)^{1/n} + (1-\lambda)\mathrm{vol}(B)^{1/n}.$$

An immediate consequence of the first theorem above is that any one-dimensional marginal distribution of a convex body is logconcave; the second says that it is in fact $(1/(n-1))$-concave if the body is in $\mathbb{R}^n$.

**Example 1.4.5.** We give one example problem related to Bayesian inference. Suppose we have a signal $\theta = (\theta_1, \theta_2, \cdots, \theta_n)$ and that we can take a measurement $y_i$ of $\theta_i$. The measurement only incurs unbiased Gaussian noise, i.e., $y_i = \theta_i + \epsilon_i$ where $\epsilon_i \sim N(0,1)$. The question is to recover the signal $\theta$ using $y$. Without any prior on $\theta$, the only sensible recovery is $\theta = y$. With a prior, one can apply Bayes' theorem

$$\mathbf{P}(\theta|y) = \frac{\mathbf{P}(y|\theta)\mathbf{P}(\theta)}{\mathbf{P}(y)}.$$

The Bayesian choice is to find $\theta$ with maximum likelihood, namely $\theta = \mathrm{argmin}_\theta - \log \mathbf{P}(\theta|y)$.

Using the noise assumption, we have that

$$\mathbf{P}(y|\theta) \propto \exp(-\frac{1}{2}\sum_i (y_i - \theta_i)^2).$$

Now, say we know the signal is smooth and model the prior as $\mathbf{P}(\theta) \propto \exp(-\lambda \sum_i (\theta_i - \theta_{i+1})^2)$ where $\lambda$ controls how smooth the signal is. Hence,

$$-\log \mathbf{P}(\theta|y) = c + \sum_i (y_i - \theta_i)^2 + \lambda \sum_i (\theta_i - \theta_{i+1})^2.$$

Since each term in the function above is convex, so is the whole formula. Hence, the recovery question becomes a convex optimization problem

$$\min_\theta \sum_i (y_i - \theta_i)^2 + \lambda \sum_i (\theta_i - \theta_{i+1})^2.$$

When we recover a signal, we want to know how confident we are because there are multiple $\theta$ that could explain the same measurement $y$. One way to do this is to sample multiple $\theta \propto \mathbf{P}(\theta|y)$ and compute the empirical variance or other statistics. Note that

$$\mathbf{P}(\theta|y) \propto e^{-\sum_i (y_i - \theta_i)^2 - \lambda \sum_i (\theta_i - \theta_{i+1})^2}$$

which is a logconcave distribution. Therefore, one can study the signal and quality of signal recovery via logconcave sampling.

## ■ 1.5 Gradient Descent

Perhaps the most natural algorithm for optimization is gradient descent. In fact it has many variants with different guarantees. Assume that the function $f$ to be optimized is continuously differentiable. By basic calculus, either the minimum is unbounded or the gradient is zero at a minimum. So we try to find a point with gradient close to zero (which, of course, does not guarantee global optimality). The basic algorithm is the following:

**Algorithm 1:** `GradientDescent` (GD)

```
Input: Initial point $x^{(0)} \in \mathbb{R}^n$, step size $h > 0$.
for $k = 0, 1, \cdots$ do
    if $\|\nabla f(x^{(k)})\|_2 \leq \epsilon$ then return $x^{(k)}$
    // Alternatively, one can use $x^{(k+1)} \leftarrow \texttt{argmin}_{x = x^{(k)} + t\nabla f(x^{(k)})} f(x)$.
    $x^{(k+1)} \leftarrow x^{(k)} - h \cdot \nabla f(x^{(k)})$.
end
```

One can view gradient descent as a greedy method for solving $\min_{x \in \mathbb{R}^n} f(x)$. At the point $x$, the gradient descent goes to the minimizer of

$$\min_{\|\Delta\|_2 \leq h \|\nabla f(x)\|_2} f(x) + \nabla f(x)^\top \Delta$$

where the term $f(x) + \nabla f(x)^\top \Delta$ is simply the first-order approximation of $f(x+\Delta)$. Note that in this problem, the current point $x$ is fixed and we are optimizing the step $\Delta$. Certainly, there is no inherent reason for using first-order approximation and the Euclidean norm $\|x\|_2$. For example, if you use second-order approximation, then you would recover the (regularized) Newton method.

If the iteration stops, we get a point with $\|\nabla f(x)\| \leq \epsilon$. Why is this good? The hope is that $x$ is a near-minimum in the neighborhood of $x$. However, this might not be true if the gradient can fluctuate wildly, i.e., the Hessian $\nabla^2 f$ can be very large (recall that the Hessian of a function is the matrix of its second derivatives). So it is natural to assume the Hessian is bounded, i.e.,

$$\left\|\nabla^2 f\right\|_{\mathrm{op}} \leq L$$

or (essentially equivalently, but without assuming twice differentiability), that $\nabla f$ is $L$-Lipschitz.

The advantage of the alternative version in the algorithm with the line search is that it does not need to explicitly know the bound $L$ on the Hessian.

## Analysis for general functions

We can prove the following guarantee.

**Theorem 1.5.1.** *Let $f$ be a function with $L$-Lipschitz gradient and $x^*$ be any minimizer of $f$. The* `GradientDescent` *with step size $h = \frac{1}{L}$ outputs a point $x$ such that $\|\nabla f(x)\| \leq \epsilon$ in $\frac{2L}{\epsilon^2}\left(f(x^{(0)}) - f(x^*)\right)$ iterations.*

The point found is only an approximate "local" minimum. The proof idea involves showing the function value $f(x)$ decrease by at least $\frac{\epsilon^2}{2L}$ when $\|\nabla f(x)\| \geq \epsilon$. Since the function value can only decrease by at most $f(x^{(0)}) - f(x^*)$, this lower bounds the number of iterations.

In the proof, we need the following 2nd order Taylor expansion.

**Lemma 1.5.2.** *For any $f \in \mathcal{C}^2(\mathbb{R}^n)$, and any $x, y \in \mathbb{R}^n$, there is a $z \in [x, y]$ s.t.*

$$f(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{1}{2}(y - x)^\top \nabla^2 f(z)(y - x).$$

*Proof.* Let $g(t) = f((1-t)x + ty)$. Taylor expansion (Theorem 0.0.1) shows that

$$g(1) = g(0) + g'(0) + \frac{1}{2} g''(\zeta)$$

where $\zeta \in [0, 1]$. Note that $g(0) = f(x)$, $g'(0) = \nabla f(x)^\top (y - x)$ and $g''(\zeta) = (y - x)^\top \nabla^2 f((1-\zeta)x + \zeta y)(y - x)$. Hence, we have the result. $\square$

Using this result, we can bound the progress for each step in the non-convex case.

**Lemma 1.5.3.** *For any $f \in \mathcal{C}^2(\mathbb{R}^n)$ with $L$-Lipschitz gradient, we have*

$$f(x - \frac{1}{L}\nabla f(x)) \leq f(x) - \frac{1}{2L}\|\nabla f(x)\|_2^2.$$

*Proof.* Lemma 1.5.2 shows that

$$f(x - \frac{1}{L}\nabla f(x)) = f(x) - \frac{1}{L}\|\nabla f(x)\|_2^2 + \frac{1}{2L^2}\nabla f(x)^\top \nabla^2 f(z) \nabla f(x)$$

for some $z \in [x, x - \frac{1}{L}\nabla f(x)]$. Since $\|\nabla^2 f(x)\|_{\mathrm{op}} \leq L$, we have that $\nabla f(x)^\top \nabla^2 f(z)\nabla f(x) \leq L \cdot \|\nabla f(x)\|_2^2$. Hence, we have the result. □

*Proof of Theorem 1.5.1.* Since each step of gradient descent decreases $f$ by $\frac{\epsilon^2}{2L}$ and since we can decrease $f$ by at most $f(x^{(0)}) - f^*$, we have the result. □

Despite the simplicity of the algorithm and the proof, it is recently shown that this is the best one can do via any algorithm for this general setting [11].

## Analysis for convex functions

Assuming the function is convex, we can prove that gradient descent in fact converges to the global minimum. In particular, when $\|\nabla f(x)\|_2$ is small, convexity shows that $f(x) - f^*$ is small (Theorem 1.2.3).

**Lemma 1.5.4.** *For any convex $f \in \mathcal{C}^1(\mathbb{R}^n)$, we have that $f(x) - f(y) \leq \|\nabla f(x)\|_2 \cdot \|x - y\|_2$ for all $x, y$.*

*Proof.* Theorem 1.2.3 and Cauchy-Schwarz inequality shows that

$$f(x) - f(y) \leq \langle \nabla f(x), x - y\rangle \leq \|\nabla f(x)\|_2 \cdot \|x - y\|_2.$$

□

This in turns give a better bound on the number of iterations because the bound in Theorem 1.5.1 is affected by $f(x) - f^*$.

**Theorem 1.5.5.** *Let $f \in \mathcal{C}^2(\mathbb{R}^n)$ be convex with $L$-Lipschitz gradient and $x^*$ be any minimizer of $f$. With step size $h = \frac{1}{L}$, the sequence $x^{(k)}$ in* `GradientDescent` *satisfies*

$$f(x^{(k)}) - f(x^*) \leq \frac{2LR^2}{k+4} \quad \text{where } R = \max_{f(x) \leq f(x^{(0)})} \|x - x^*\|_2.$$

*Proof.* Let $\epsilon_k = f(x^{(k)}) - f(x^*)$. Lemma 1.5.3 shows that

$$f(x^{(k+1)}) = f(x^{(k)} - \frac{1}{L}\nabla f(x^{(k)})) \leq f(x^{(k)}) - \frac{1}{2L}\|\nabla f(x^{(k)})\|_2^2.$$

Subtracting both sides by $f(x^*)$, we have $\epsilon_{k+1} \leq \epsilon_k - \frac{1}{2L}\|\nabla f(x^{(k)})\|_2^2$. Lemma 1.5.4 shows that

$$\epsilon_k \leq \|\nabla f(x^{(k)})\|_2 \cdot \|x^{(k)} - x^*\|_2 \leq \|\nabla f(x^{(k)})\|_2 \cdot R.$$

Therefore, we have that

$$\epsilon_{k+1} \leq \epsilon_k - \frac{1}{2L}\left(\frac{\epsilon_k}{R}\right)^2.$$

Also, we have that

$$\epsilon_0 = f(x^{(0)}) - f^* \leq \nabla f(x^*)^\top (x^{(0)} - x^*) + \frac{L}{2}\|x^{(0)} - x^*\|^2 \leq \frac{LR^2}{2}.$$

Now, we need to solve the recursion. We note that

$$\frac{1}{\epsilon_{k+1}} - \frac{1}{\epsilon_k} = \frac{\epsilon_k - \epsilon_{k+1}}{\epsilon_k \epsilon_{k+1}} \geq \frac{\epsilon_k - \epsilon_{k+1}}{\epsilon_k^2} \geq \frac{1}{2LR^2}.$$

Therefore, after $k$ iterations, we have

$$\frac{1}{\epsilon_k} \geq \frac{1}{\epsilon_0} + \frac{k}{2LR^2} \geq \frac{2}{LR^2} + \frac{k}{2LR^2} = \frac{k+4}{2LR^2}.$$

$\square$

*Remark* 1.5.6. This proof is a typical in optimization. It shows that when the gradient is large, then we make large progress and when the gradient is small, we are close to optimal.

*Remark* 1.5.7. This proof did not use any property of $\ell_2$ or inner product space. Therefore, it works for general norms if the gradient descent step is defined using that norm. For the case of $\ell_2$, one can prove that $\|x^{(k)} - x^*\|$ is in fact decreasing and hence the final bound is simply $\frac{2L\|x^{(0)}-x^*\|^2}{k+4}$.

*Remark* 1.5.8. Rewriting the bound, Theorem 1.5.5 shows it takes $\frac{2L\|x^{(0)}-x^*\|^2}{\epsilon}$ iterations. Compare to the bound $\frac{2L}{\epsilon^2}\left(f(x^{(0)}) - f^*\right)$ in Theorem 1.5.1, it seems the new result has a strictly better dependence on $\epsilon$. However, this is not true because one is measuring the error in terms of $\|\nabla f(x)\|$ and one is measuring the error in terms of $f(x) - f^*$. For $f(x) = x^2$, we have $f(x) - f^* = \|\nabla f(x)\|^2$ and hence both have the same dependence on $\epsilon$ for this particular function. In general, the $\epsilon$ dependence of both bounds is the same asymptotically. The real benefit of Theorem 1.5.5 is its global convergence.

## 1.6 Strongly Convex Functions

We note that the convergence rate $\epsilon^{-1}$ or $\epsilon^{-2}$ is awful if we need to solve the problem up to machine accuracy. Getting the machine accuracy is sometimes important if the optimization problem is used as a subroutine. We note that for the case $f(x) = \frac{1}{2}\|x\|^2$, the gradient descent with step size $h = 1$ takes exactly 1 step. Therefore, it is natural to ask if one can improve the bound for functions close to quadratics. This motivates the following assumption:

**Definition 1.6.1.** We call a function $f \in \mathcal{C}^1(\mathbb{R}^n)$ is $\mu$-strongly convex if for any $x, y \in \mathbb{R}^n$

$$f(y) \geq f(x) + \nabla f(x)^\top (y-x) + \frac{\mu}{2}\|y-x\|^2.$$

*Remark* 1.6.2. When $\mu = 0$, this gives an alternative definition of convex function for $\mathcal{C}^1$ functions. We left the proof of the equivalence between two definitions of convex functions as an exercise.

Alternatively, one can understand the convexity and $\mu$-strong convexity by $\nabla^2 f(x)$.

**Theorem 1.6.3.** *Let $f \in \mathcal{C}^2(\mathbb{R}^n)$. Then, $f$ is $\mu$-strongly convex if and only if*

$$\nabla^2 f(x) \succeq \mu \cdot I \text{ for all } x \in \mathbb{R}^n.$$

*Proof.* Suppose that $\nabla^2 f(x) \succeq \mu \cdot I$ for all $x \in \mathbb{R}^n$. For any $x, y \in \mathbb{R}^n$, Lemma 1.5.2 shows that

$$\begin{aligned} f(y) &= f(x) + \nabla f(x)^\top (y-x) + \frac{1}{2}(y-x)^\top \nabla^2 f(z)(y-x) \\ &\geq f(x) + \nabla f(x)^\top (y-x) + \frac{\mu}{2}\|y-x\|_2^2 \end{aligned}$$

where we used $\nabla^2 f(z) \succeq \mu \cdot I$ at the end. This shows that $f$ is $\mu$-strongly convex.

If $f$ is $\mu$-strongly convex, for any $x, h \in \mathbb{R}^n$

$$f(x+th) \geq f(x) + t\nabla f(x)^\top h + \frac{\mu t^2}{2}\|h\|_2^2.$$

By Lemma 1.5.2, we have that

$$f(x+th) = f(x) + t\nabla f(x)^\top h + \frac{t^2}{2} h^\top \nabla^2 f(z) h$$

where $z \in [x, x+th]$. By comparing two equations, we have that $h^\top \nabla^2 f(z) h \geq \mu \|h\|_2^2$. Taking $t \to 0$, we have $z \to x$ and hence $\nabla^2 f(z) \to \nabla^2 f(x)$. Therefore, we have that

$$h^\top \nabla^2 f(x) h \geq \mu \|h\|_2^2$$

for all $x$ and $h$. Hence, this gives $\nabla^2 f(x) \succeq \mu \cdot I$ for all $x$. □

Now, we study gradient descent for $\mu$-strongly convex functions.

**Theorem 1.6.4.** *Let $f \in \mathcal{C}^2(\mathbb{R}^n)$ be $\mu$-strongly convex with $L$-Lipschitz gradient and $x^*$ be any minimizer of $f$. With step size $h = \frac{1}{L}$, the sequence $x^{(k)}$ in* `GradientDescent` *satisfies*

$$f(x^{(k)}) - f(x^*) \leq (1 - \frac{\mu}{L})^k (f(x^{(0)}) - f(x^*)).$$

In a later chapter, we will see that an *accelerated* variant of gradient descent improves this further by replacing the $\frac{\mu}{L}$ term with $\sqrt{\frac{\mu}{L}}$.

*Proof.* Lemma 1.5.3 shows that

$$f(x^{(k+1)}) - f^* \leq f(x^{(k)}) - f^* - \frac{1}{2L}\|\nabla f(x^{(k)})\|_2^2. \tag{1.6.1}$$

Next, the definition of $\mu$ strongly convex showed that

$$f(x^*) \geq f(x^{(k)}) + \nabla f(x^{(k)})^\top (x^* - x^{(k)}) + \frac{\mu}{2}\|x^* - x^{(k)}\|_2^2.$$

Rearranging the term, we have

$$f(x^{(k)}) - f^* \leq \nabla f(x^{(k)})^\top (x^{(k)} - x^*) - \frac{\mu}{2}\|x^{(k)} - x^*\|_2^2 \leq \max_{\Delta} \left( \nabla f(x^{(k)})^\top \Delta - \frac{\mu}{2}\|\Delta\|_2^2 \right) = \frac{1}{2\mu}\|\nabla f(x^{(k)})\|_2^2. \tag{1.6.2}$$

Putting this into the gradient term in (1.6.1) gives

$$f(x^{(k+1)}) - f^* \leq (1 - \frac{\mu}{L})(f(x^{(k)}) - f^*).$$

The conclusion follows. □

## Discussion

In continuous time, gradient descent follows the ODE

$$\frac{dx_t}{dt} = -\nabla f(x_t).$$

This can be viewed as the canonical continuous algorithm. Finding the right discretization has lead to many fruitful research directions. One benefit of the continuous view is to simplify some calculation. For example, the Theorem 1.6.4 now becomes

$$\frac{d}{dt}(f(x_t) - f(x^*)) = \nabla f(x_t)^\top \frac{dx_t}{dt} = -\|\nabla f(x_t)\|^2 \leq -2\mu(f(x_t) - f(x^*))$$

where we used (1.6.2) at the end. Solving this differential inequality, we have

$$f(x_t) - f(x^*) \leq e^{-2\mu t}(f(x_0) - f(x^*)).$$

However, we emphasize that this continuous view is mainly useful for getting a simple result instead of a tight result.

On a separate note, it is natural to ask to what extent the assumption of convexity is essential for the bounds we obtained. This is the motivation for the next exercises.

**Exercise 1.6.5.** Suppose $f$ satisfies $\langle \nabla f(x), x - x^* \rangle \geq \alpha \left( f(x) - f(x^*) \right)$. Derive a bound similar to Theorem 1.5.5 for gradient descent.

**Exercise 1.6.6.** Suppose $f$ satisfies $\|\nabla f(x)\|^2 \geq \mu\left(f(x) - f(x^*)\right)$. Derive a bound similar to Theorem 1.6.4 for gradient descent.

**Exercise 1.6.7.** Give examples of nonconvex functions satisfying the above conditions. (Note: convex functions satisfy the first with $\mu = 1$ and $\mu$-strongly convex functions satisfy the second.)

## 1.7 Langevin Dynamics

Here we study a simple stochastic process for generating samples from a desired distribution $e^{-f(x)}$. It can also be viewed as a stochastic version of gradient descent. While gradient descent corresponds to an ordinary differential equation (ODE), stochastic gradient descent corresponds to a stochastic differential equation (SDE).

**Algorithm 2:** `LangevinDynamics` (LD)

```
Input: Initial point $x_0 \in \mathbb{R}^n$.
Solve the stochastic differential equation
    $dx_t = -\nabla f(x_t)dt + \sqrt{2}dW_t$
Output: $x_t$.
```

Here $f : \mathbb{R}^n \to \mathbb{R}$ is a function, $x_t$ is the random variable at time $t$ and $dW_t$ is infinitesimal Brownian motion also known as a Wiener process. While it takes some care to define rigorously, for now we can view it as the following discrete process

$$x_{t+h} = x_t - h\nabla f(x_t) + \sqrt{2h}\zeta_t$$

with $\zeta_t$ sampled independently from $N(0, I)$. When we take the step size $h \to 0$, this discrete process converges to the continuous one. We choose to discuss the continuous version here only for simplicity. First, we show that this process converges to $e^{-f}$ in continuous time. The proof relies on the following general theorem about the distribution induced by an SDE.

**Theorem 1.7.1** (Fokker–Planck equation*)**.** *For any process $x_t \in \mathbb{R}^n$ satisfying $dx_t = \mu(x_t)dt + \sigma(x_t)dW_t$ where $\mu(x_t) \in \mathbb{R}^n$ and $\sigma(x_t) \in \mathbb{R}^{n\times m}$ with the initial point $x_0$ drawn from $p_0$. Then, the distribution $p_t$ of $x_t$ satisfies the equation*

$$\frac{dp_t}{dt} = -\sum_i \frac{\partial}{\partial x_i}(\mu(x)_i p_t(x)) + \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\left[(D(x))_{ij} p_t(x)\right]$$

*where $D(x) = \sigma(x)\sigma(x)^\top$.*

*Proof.* For any smooth function $\phi$, we have that

$$\mathbf{E}_{x\sim p_t}\phi(x) = \mathbf{E}\phi(x_t).$$

Taking derivatives on the both sides with respect to $t$ and using Itô's lemma (Lemma 0.0.3), we have that

$$\begin{aligned}\int \phi(x)dp_t(x)dx &= \mathbf{E}\left(\nabla\phi(x_t)^\top\mu(x_t)dt + \nabla\phi(x_t)^\top\sigma(x_t)dW_t + \frac{1}{2}\mathrm{tr}(\sigma(x_t)^\top\nabla^2\phi(x_t)\sigma(x_t))dt\right)\\ &= \mathbf{E}\left(\nabla\phi(x_t)^\top\mu(x_t)dt + \frac{1}{2}\mathrm{tr}(\nabla^2\phi(x_t)D(x_t))dt\right).\end{aligned}$$

Using $x_t \sim p_t$, we have that

$$\int \phi(x)\frac{dp_t}{dt}dx = \int \nabla\phi(x)^\top\mu(x)p_t(x) + \frac{1}{2}\mathrm{tr}(\nabla^2\phi(x)D(x))p_t(x)dx.$$

Integrating by parts,

$$\int \nabla\phi(x)^\top\mu(x)p_t(x)dx = -\int \phi(x)\sum_i\frac{\partial}{\partial x_i}(\mu_i(x)p_t(x))dx.$$

Similarly, integrating by parts twice gives

$$\begin{aligned}\int \operatorname{tr}(\sigma(x)^\top \nabla^2\phi(x)\sigma(x))p_t(x)dx &= \int \operatorname{tr}(\nabla^2\phi(x)\sigma(x)\sigma(x)^\top)p_t(x)dx \\ &= \sum_{i,j}\int \phi(x)\frac{\partial^2}{\partial x_i \partial x_j}\left[(D(x))_{ij}p_t(x)\right]dx.\end{aligned}$$

Hence,

$$\int \phi(x)\left[\frac{dp_t}{dt} + \sum_i \frac{\partial}{\partial x_i}(\mu(x)_i p_t(x)) - \frac{1}{2}\sum_{i,j}\frac{\partial^2}{\partial x_i \partial x_j}\left[(D(x))_{ij}p_t(x)\right]\right]dx = 0$$

for any smooth $\phi$. Therefore, we have the conclusion of the lemma.

We apply the Fokker–Planck equation to the Langevin dynamics. □

**Theorem 1.7.2.** *For any smooth function $f$, the density proportional to $F = e^{-f}$ is stationary for the Langevin dynamics.*

*Proof.* The Fokker–Planck equation (Theorem 1.7.1) shows that the distribution $p_t$ of $x_t$ satisfies

$$\frac{dp_t}{dt} = \sum_i \frac{\partial}{\partial x_i}\left(\frac{\partial f(x)}{\partial x_i}p_t(x)\right) + \sum_i \frac{\partial^2}{\partial x_i^2}\left[p_t(x)\right]. \tag{1.7.1}$$

We can verify that $p_t(x) \propto e^{-f(x)}$ is a solution. □

**Convergence via Coupling.** Next we turn to the rate of convergence, which will also prove uniqueness. For this, we assume that $f$ is strongly convex. The proof is via the classical coupling technique (cite: Aldous). Briefly, two distributions are compared via some probabilistic distance. For example, the total variation distance can be bounded by the probability that random variables from the two distributions are not identical, under any matching (assignment) between the two distributions. We then couple two copies $x_t, y_t$ of the process with different starting points (the coupling is a joint distribution $D(x_t, y_t)$ with the property that its marginal for each of $x_t, y_t$ is exactly the process) and show that their distributions get closer over time. While the challenge is usually to find a good coupling, in the present case, the simplest identity coupling works (for the continuous time convergence).

**Lemma 1.7.3.** *Let $x_t, y_t$ evolves according to the Langevin diffusion for a $\mu$-strongly convex function $f : \mathbb{R}^n \to \mathbb{R}$. Then, there is a coupling between $x_t, y_t$ s.t.*

$$\mathbf{E}\left\|x_t - y_t\right\| \le e^{-\mu t}\left\|x_0 - y_0\right\|.$$

*Proof.* From the definition of LD, and by using the same Gaussian $dW_t$ for both process, we have that

$$\frac{d}{dt}(x_t - y_t) = \nabla f(y_t) - \nabla f(x_t).$$

Hence,

$$\frac{1}{2}\frac{d}{dt}\|x_t - y_t\|^2 = 2\left\langle \nabla f(y_t) - \nabla f(x_t), x_t - y_t\right\rangle.$$

Next, from the strong convexity of $f$, we have

$$\begin{aligned} f(y_t) - f(x_t) &\ge \nabla f(x_t)^\top(y_t - x_t) + \frac{\mu}{2}\left\|x_t - y_t\right\|^2, \\ f(x_t) - f(y_t) &\ge \nabla f(y_t)^\top(x_t - y_t) + \frac{\mu}{2}\left\|x_t - y_t\right\|^2. \end{aligned}$$

Adding two equations together, we have

$$(\nabla f(x_t) - \nabla f(y_t))^\top(x_t - y_t) \ge \mu\left\|x_t - y_t\right\|^2.$$

Therefore,

$$\frac{1}{2}\frac{d}{dt}\|x_t - y_t\|^2 \le -\mu\|x_t - y_t\|^2.$$

Hence,

$$\frac{d}{dt}\log\|x_t-y_t\|^2=\frac{\frac{d}{dt}\|x_t-y_t\|^2}{\|x_t-y_t\|^2}\leq -2\mu.$$

Integrating both sides from 0 to $t$, we get

$$\log\|x_t-y_t\|^2\leq\log\|x_0-y_0\|^2-2\mu t$$

which proves the result. □

## 1.8 Langevin Dynamics is Gradient Descent in Density Space*[2]

Here we show that Langevin dynamics is simply gradient descent for the function $F(\rho)=D_{\mathrm{KL}}(\rho\|\nu)$ on the Wasserstein space where $\nu=e^{-f(x)}/\int e^{-f(y)}dy$. For this, we first define the Wasserstein space.

**Definition 1.8.1.** The Wasserstein space $P_2(\mathbb{R}^n)$ on $\mathbb{R}^n$ is the manifold on the set of probability measures on $\mathbb{R}^n$ such that the shortest path distance of two measures $x,y$ in this manifold is exactly equal to the Wasserstein distance between $x$ and $y$.

**Lemma 1.8.2.** *For any $p\in P_2(\mathbb{R}^n)$ and $v\in T_pP_2(\mathbb{R}^n)$, we can write $v(x)=\nabla\cdot(p(x)\nabla\lambda(x))$ for some function $\lambda$ on $\mathbb{R}^n$. Furthermore, the length of $v$ in this metric is given by*

$$\|v\|_p^2=\mathbf{E}_{x\sim p}\|\nabla\lambda(x)\|^2.$$

*Proof.* Let $p\in P_2(\mathbb{R}^n)$ and $v\in T_pP_2(\mathbb{R}^n)$. We will show that any change of density $v$ can be represented by a vector field $c$ on $\mathbb{R}^n$ as follows: Consider the process $x_0\sim p$ and $\frac{d}{dt}x_t=c(x_t)$. Let $p_t$ be the density of the distribution of $x_t$. To compute $\frac{d}{dt}p_t$, we follow the same idea as in the proof as Theorem 1.7.1. For any smooth function $\phi$, we have that $\mathbf{E}_{x\sim p_t}\phi(x)=\mathbf{E}\phi(x_t)$. Taking derivatives on the both sides with respect to $t$, we have that

$$\int\phi(x)\frac{d}{dt}p_t(x)dx=\int\nabla\phi(x)^\top c(x)p_t(x)dx=-\int\nabla\cdot(c(x)p_t(x))\phi(x)dx$$

where we used integration by parts at the end. Since this holds for all $\phi$, we have that

$$\frac{dp_t(x)}{dt}=-\nabla\cdot(p_t(x)c(x)).$$

Since we are interested only inthe vector fields that generate minimum movement in Wasserstein distance, we consider the optimization problem

$$\min_{-\nabla\cdot(pc)=v}\frac{1}{2}\int p(x)\|c(x)\|^2dx$$

where we can think $v$ is the change of $p_t$. Let $\lambda(x)$ be the Lagrangian multiplier of the constraint $-\nabla\cdot(pc)=v$. Then, the problem becomes

$$\begin{aligned}&\min_c\frac{1}{2}\int p(x)\|c(x)\|^2dx-\int\lambda(x)\nabla\cdot(p(x)c(x))dx.\\=&\min_c\frac{1}{2}\int p(x)\|c(x)\|^2dx+\int\nabla\lambda(x)^\top c(x)\cdot p(x)dx.\end{aligned}$$

Now, we note that the problem is a pointwise optimization problem with the minimizer is given by

$$c(x)=-\nabla\lambda(x).$$

This proves that any vector fields that generate minimum movement in Wasserstein distance is a gradient field. Also, we have that $v(x)=\nabla\cdot(p(x)\nabla\lambda(x))$. Note that the right hand side is an elliptical differential equation and hence for any $v$ with $\int v(x)dx=0$, there is an unique solution $\lambda(x)$. Therefore, we can write $v(x)=\nabla\cdot(p(x)\nabla\lambda(x))$ for some $\lambda(x)$.

[2]Anything marked with * means it is probably too mathematical and could be skipped.

Next, we note that the movement is given by

$$\|v\|_p^2 = \int p(x)\|c(x)\|^2 dx = \mathbf{E}_{x\sim p}\|\nabla\lambda(x)\|^2.$$

□

As we discussed in the gradient descent section, one can use norms other than $\ell_2$ norm. For the Wasserstein space, we should use the local norm.

**Theorem 1.8.3.** *Let $\rho_t$ be the density of the distribution produced by Langevin Dynamics for the target distribution $\nu = e^{-f(x)}/\int e^{-f(y)}dy$. Then, we have that*

$$\frac{d\rho}{dt} = \mathrm{argmin}_{v\in T_p P_2(\mathbb{R}^n)} \langle \nabla F(\rho), v\rangle_p + \frac{1}{2}\|v\|_p^2.$$

*Namely, $\rho_t$ follows continuous gradient descent in the density space for the function $F(\rho) = D_{\mathrm{KL}}(\rho\|\nu)$ under the Wasserstein metric.*

*Proof.* For any function $c$, the optimization problem of interest satisfies

$$\min_{\delta=\nabla\cdot(\rho\nabla\lambda)} \langle c,\delta\rangle + \frac{1}{2}\int \rho(x)\|\nabla\lambda(x)\|^2 dx = \min_{\nabla\lambda} -\int \rho(x)\cdot\nabla c(x)^\top\nabla\lambda(x)dx + \frac{1}{2}\int\rho(x)\|\nabla\lambda(x)\|^2 dx.$$

Solving the right hand side, we have $\nabla c = \nabla\lambda$ and hence $\delta = \nabla\cdot(\rho\nabla c)$. Now, we note that $\nabla F(\rho) = \log\frac{\rho}{\nu} - 1$. Therefore,

$$\begin{aligned}\frac{d\rho}{dt} &= \nabla\cdot(\rho\nabla(\log\frac{\rho}{\nu} - 1))\\ &= \nabla\cdot(\rho\nabla\log\frac{\rho}{\nu})\\ &= \nabla\cdot(\rho\nabla f) + \Delta\rho\end{aligned}$$

which is exactly equal to (1.7.1). □

To analyze this continuous descent on the Wasserstein space, we first prove that continuous gradient descent converges exponentially whenever $F$ is strongly convex.

**Lemma 1.8.4.** *Let $F$ be a function satisfying "Gradient Dominance":*

$$\|\nabla F(x)\|_x^2 \geq \alpha\cdot(F(x) - \min_y F(y)) \quad \textit{for all } x \tag{1.8.1}$$

*on the manifold with the metric $\|\cdot\|_x$ where $\nabla$ is the gradient on the manifold. Then, the process $dx_t = -\nabla F(x_t)dt$ converges exponentially, i.e., $F(x_t) - \min_y F(y) \leq e^{-\alpha t}(F(x_0) - \min_y F(y))$.*

*Proof.* We write

$$\frac{d}{dt}(F(x) - \min_y F(y)) = \langle\nabla F(x_t), \frac{dx_t}{dt}\rangle_{x_t} = -\|\nabla F(x_t)\|_{x_t}^2 \leq -\alpha(F(x) - \min_y F(y)).$$

The conclusion follows. □

Finally, we note that the log-Sobolev inequality for the density $\nu$ can be re-stated as the condition (1.8.1).

**Lemma 1.8.5.** *Fix a density $\nu$. Then the log-Sobolev inequality, namely, for every smooth function $g$,*

$$2\int \|\nabla g\|^2\ d\nu \geq \alpha\int g(x)^2\log g(x)^2\ d\nu$$

*implies the condition (1.8.1).*

*Proof.* Take $g(x) = \sqrt{\frac{\rho(x)}{\nu(x)}}$, the log-Sobolev inequality shows that

$$\frac{1}{2}\int \rho(x)\left\|\nabla \log \frac{\rho(x)}{\nu(x)}\right\|^2 dx \geq \alpha \cdot \int \rho(x) \log \frac{\rho(x)}{\nu(x)} dx \text{ for all } \rho.$$

As we calculate in Theorem 1.8.3, we have that

$$\|\nabla F(\rho)\|_\rho^2 = \int \rho(x)\left\|\nabla \log \frac{\rho(x)}{\nu(x)}\right\|^2 dx.$$

Therefore, this is exactly the condition (1.8.1) with coefficient $2\alpha$. □

Combining Lemma 1.8.5 and Lemma 1.8.4, we have the following result:

**Theorem 1.8.6.** *Let $f$ be a smooth function with log-Sobolev constant $\alpha$. Then the Langevin dynamics*

$$dx_t = -\nabla f(x)dt + \sqrt{2}dW_t$$

*converges exponentially in KL-divergence to the density $\nu(x) \propto e^{-f(x)}$ with mixing rate $O(\frac{1}{\alpha})$, i.e., $KL(x_t,\nu) \leq e^{-2\alpha t}KL(x_0,\nu)$.*

See [35] for a tight estimate of log-Sobolev constant for logconcave measures. In particular for a logconcave measure with support of diameter $D$, the log-Sobolev constant is $\Omega(1/D)$.

## Discussion

Langevin dynamics samples from strongly logconcave densities, and the convergence is fast in continuous time. Turning this into an efficient algorithm takes more work and assumptions. As we saw in Section 1.8, it turns out that Langevin dynamics is in fact gradient descent in the space of probability measures under the Wasserstein metric, where the function being minimized is the KL-divergence of the current density from the target stationary density. For more on this view, see [63].

# Chapter 2

# Elimination

## 2.1 Cutting Plane Methods

The goal of this chapter is to present some polynomial-time algorithms for convex optimization. These algorithms use the knowledge of the function in a minimal way, essentially by querying the function value and knowing some (weak) bounds on its support/value.

Given a continuously differentiable convex function $f$, Theorem 1.2.3 shows that

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \text{ for all } y. \tag{2.1.1}$$

Let $x^*$ be any minimizer of $f$. Putting $y = x^*$, we have that

$$f(x) \geq f(x^*) \geq f(x) + \langle \nabla f(x), x^* - x \rangle .$$

Therefore, we know that $\langle \nabla f(x), x^* - x \rangle \leq 0$. Namely, $x^*$ lies in a halfspace pointing away from the $\nabla f(x)$. Roughly speaking, this shows that each gradient computation cuts the set of possible solutions in half. In one dimension, this allows us to do a binary search to minimize convex functions.

It turns out that in $\mathbb{R}^n$, binary search still works. In this chapter, we will cover several ways to do this binary search. All of them follow the same framework, called the *cutting plane method.*

In this framework, we maintain a convex set $E^{(k)}$ that contains the minimizer $x^*$ of $f$. Each iteration, we compute the gradient of $f$ at a certain point $x^{(k)}$ depending on $E^{(k)}$. The convexity of $f$ shows that any minimizer $x^*$ lies in the halfspace (2.1.2) and hence $x^* \in H^{(k)} \cap E^{(k)}$. The algorithm continues by choosing $E^{(k+1)}$ which contains $H^{(k)} \cap E^{(k)}$.

**Algorithm 3:** `CuttingPlaneFramework`

```
Input: Initial set $E^{(0)} \subseteq \mathbb{R}^n$.
for $k = 0, \cdots$ do
    Find a point $x^{(k)} \in E^{(k)}$.
    if $E^{(k)}$ is "small" enough then return $x^{(k)}$.
    Find $E^{(k+1)} \supset E^{(k)} \cap H^{(k)}$ where
        $H^{(k)} \overset{\text{def}}{=} \{x \in \mathbb{R}^n \text{ such that } \nabla f(x^{(k)})^\top (x - x^{(k)}) \leq 0\}.$   (2.1.2)
end
```

The main questions are

1. How do we choose $x^{(k)}$ and $E^{(k+1)}$?
2. How do we measure progress?
3. How quickly does the method converge?
4. How expensive is each step?

Progress on the cutting plane method is shown in the next table.

| Year | $E^{(k)}$ and $x^{(k)}$ | Iter | Cost/Iter |
|---|---|---|---|
| 1965 [36, 49] | Center of gravity | $n$ | $n^n$ |
| 1979 [65, 56, 31] | Center of ellipsoid | $n^2$ | $n^2$ |
| 1988 [30] | Center of John ellipsoid | $n$ | $n^{2.878}$ |
| 1989 [60] | Volumetric center | $n$ | $n^{2.378}$ |
| 1995 [6] | Analytic center | $n$ | $n^{2.378}$ |
| 2004 [9] | Center of gravity | $n$ | $n^6$ |
| 2015 [34] | Hybrid center | $n$ | $n^2$ (amortized) |

**Table 2.1:** Different Cutting Plane Methods. Omitted polylogarithmic terms. The number of iterations follows from the rate.

## 2.2 Ellipsoid Method

We first start by explaining the simplest algorithm, the ellipsoid method. We maintain an ellipsoid

$$E^{(k)} \overset{\text{def}}{=} \{x \in \mathbb{R}^n : (x - x^{(k)})^\top A^{(k)}(x - x^{(k)}) \leq 1\}$$

that contains all minimizers of $f$. After we compute $\nabla f(x^{(k)})$ and $H^{(k)}$ via (2.1.2), we define $E^{(k+1)}$ to be the smallest volume ellipsoid containing $E^{(k)} \cap H^{(k)}$. The key observation is that the volume of the ellipsoid $E^{(k)}$ decrease by $1 - \Theta(\frac{1}{n})$ every iteration. This volume property holds for any halfspace through the center of the current ellipsoid (not only for the one whose normal is the gradient), a property we will exploit in the next chapter.

**Algorithm 4:** `Ellipsoid`

```
Input: Initial ellipsoid $E^{(0)} = \{x \in \mathbb{R}^n : (x - x^{(0)})^\top A^{(0)}(x - x^{(0)}) \leq 1\}$.
$M^{(0)} = \left(A^{(0)}\right)^{-1}$. // We maintain the inverse of $A$ and call it $M$.
for $k = 0, \cdots$ do
    if $E^{(k)}$ is "small" enough then return $x^{(k)}$.
    $\widehat{g}^{(k)} \leftarrow \frac{\nabla f(x^{(k)})}{\sqrt{\nabla f(x^{(k)})^\top M^{(k)} \nabla f(x^{(k)})}}$.
    $x^{(k+1)} \leftarrow x^{(k)} - \frac{1}{n+1} A^{(k)} \widehat{g}^{(k)}$.
    $A^{(k+1)} \leftarrow \left(1 - \frac{1}{n^2}\right)\left(A^{(k)} + \frac{2}{n-1}\widehat{g}^{(k)}(\widehat{g}^{(k)})^\top\right)$.
    $M^{(k+1)} \leftarrow \frac{n^2}{n^2-1}\left(M^{(k)} - \frac{2}{n+1} M^{(k)}\widehat{g}^{(k)}(\widehat{g}^{(k)})^\top M^{(k)}\right)$.
end
```

**Lemma 2.2.1.** *For the ellipsoid method (Algorithm 4), we have* $\mathrm{vol}E^{(k+1)} < e^{-\frac{1}{2n+2}}\mathrm{vol}E^{(k)}$.

*Remark* 2.2.2. Note that the proof below also shows that $\mathrm{vol}E^{(k+1)} = e^{-\Theta(\frac{1}{n})}\mathrm{vol}E^{(k)}$. Therefore, the ellipsoid method does not run faster for nice functions, making it provably slow in practice.

*Proof.* Note that the ratio of $\mathrm{vol}E^{(k+1)}/\mathrm{vol}E^{(k)}$ does not change under any affine transformation. Therefore, we can do a transformation so that $A^{(k)} = I$, $x^{(k)} = 0$ and $\nabla f(x^{(k)}) = e_1$. This simplified our calculation. We need to prove two statements: $\mathrm{vol}E^{(k+1)} < e^{-\frac{1}{2n+2}}\mathrm{vol}E^{(k)}$ and that $E^{(k)} \cap H^{(k)} \subset E^{(k+1)}$. In Lemma 2.2.4, we proved that $M^{(k)} = (A^{(k)})^{-1}$ for all $k$. We will use this fact throughout the proof.

Claim 1: $\mathrm{vol}E^{(k+1)} < e^{-\frac{1}{2(n+1)}}\mathrm{vol}E^{(k)}$.

Note that $\widehat{g}^{(k)} = e_1$ and hence $A^{(k+1)} = \left(1 - \frac{1}{n^2}\right)\left(I + \frac{2}{n-1}e_1e_1^\top\right)$. Therefore, we have that

$$\begin{aligned}\left(\frac{\mathrm{vol}E^{(k+1)}}{\mathrm{vol}E^{(k)}}\right)^2 = \left|\frac{\det A^{(k)}}{\det A^{(k+1)}}\right| &= \left(\frac{n^2}{n^2-1}\right)^n \det\left(I + \frac{2}{n-1}e_1e_1^\top\right)^{-1}\\ &= \left(\frac{n^2}{n^2-1}\right)^{n-1} \frac{n^2}{n^2-1}\left(\frac{n-1}{n+1}\right)\\ &= \left(1 + \frac{1}{n^2-1}\right)^{n-1}\left(1 - \frac{1}{n+1}\right)^2\\ &\leq \exp\left(\frac{n-1}{n^2-1} - \frac{2}{n+1}\right) = \exp\left(-\frac{1}{n+1}\right)\end{aligned}$$

where we used $1 + x \leq e^x$ for all $x$.

Claim 2. $E^{(k)} \cap H^{(k)} \subset E^{(k+1)}$.

By the definition of $E^{(k)}$and the assumption $A^{(k)} = I$, for any $x \in E^{(k)} \cap H^{(k)}$, we have that $\|x\|_2 \leq 1$ and $x_1 \leq 0$. By direct computation, we have

$$\begin{aligned}&\left(x + \frac{e_1}{n+1}\right)^\top \left(1 - \frac{1}{n^2}\right)\left(I + \frac{2}{n-1}e_1e_1^\top\right)\left(x + \frac{e_1}{n+1}\right)\\ =&\left(1 - \frac{1}{n^2}\right)\left(\|x\|^2 + \frac{2x_1(1+x_1)}{n-1} + \frac{1}{n^2-1}\right)\\ \leq&\left(1 - \frac{1}{n^2}\right)\left(1 + 0 + \frac{1}{n^2-1}\right) = 1\end{aligned}$$

where we used that $\|x\|_2 \leq 1$ and $x_1(1+x_1) \leq 0$ (because $-1 \leq x_1 \leq 0$) at the end. This shows that $x \in E^{(k+1)}$. ☐

We note that the ellipsoid method can be written without using $M^{(k)}$. In this case, the formula becomes $\widehat{g}^{(k)} \leftarrow \frac{\nabla f(x^{(k)})}{\sqrt{\nabla f(x^{(k)})^\top (A^{(k)})^{-1}\nabla f(x^{(k)})}}$. In general, it takes $O(n^{2.38\cdots})$ time to invert a matrix and this is the most expensive step of the algorithm. To avoid this bottleneck, we maintain the inverse of $A^{(k)}$ and call it $M^{(k)}$. The update rule comes from the following formula:

**Lemma 2.2.3** (Sherman-Morrison formula)**.** *For any invertible $A$ such that $1 + v^\top A^{-1}u \neq 0$, we have that*

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top A^{-1}u}.$$

In more general, many matrix function $f(A)$ satisfy the property that $f(A + uu^\top) - f(A)$ is approximately low rank. To formally see it, one can use the Cauchy integral formula [58].

**Lemma 2.2.4.** *Each step of ellipsoid method takes $O(n^2)$ time. Furthermore, we have that $M^{(k)} = (A^{(k)})^{-1}$ for all $k$.*

*Proof.* Instead of computing $(A^{(k)})^{-1}$ from scratch, we store the matrix $(A^{(k)})^{-1}$ directly and Lemma 2.2.3 shows that this can be updated by

$$\begin{aligned}(A^{(k+1)})^{-1} &= \frac{n^2}{n^2-1}\left((A^{(k)})^{-1} - \frac{2}{n-1}\frac{(A^{(k)})^{-1}\widehat{g}^{(k)}(\widehat{g}^{(k)})^\top(A^{(k)})^{-1}}{1 + \frac{2}{n-1}(\widehat{g}^{(k)})^\top(A^{(k)})^{-1}\widehat{g}^{(k)}}\right)\\ &= \frac{n^2}{n^2-1}\left((A^{(k)})^{-1} - \frac{2}{n+1}(A^{(k)})^{-1}\widehat{g}^{(k)}(\widehat{g}^{(k)})^\top(A^{(k)})^{-1}\right)\\ &= \frac{n^2}{n^2-1}\left(M^{(k)} - \frac{2}{n+1}M^{(k)}\widehat{g}^{(k)}(\widehat{g}^{(k)})^\top M^{(k)}\right).\end{aligned}$$

Hence, induction shows that $M^{(k)} = (A^{(k)})^{-1}$ for all $k$. Finally, we note that this formula can be implemented in $O(n^2)$ time. ☐

**Exercise 2.2.5.** Show that the ellipsoid $E^{(k+1)}$ computed above is the minimum volume ellipsoid containing $E^{(k)} \cap H^{(k)}$.

## ■ 2.3 From Volume to Function Value

Lemma 2.2.1 shows that the volume of a set containing all minimizers decreases by a constant factor every $n$ steps. In general, knowing that the optimal $x^*$ lies in a small volume set does not provide enough information to find a point with small function value. For example, if we only knew that $x^*$ lies in the plane $\{x : x_1 = 0\}$, we still need to search for $x^*$ over an $n-1$ dimensional space. However, if the set is constructed by the cutting plane framework (Algorithm 3), then we can guarantee that small volume implies that any point in the set has close to optimal function value.

This in turns implies that we can minimize any convex function with $\varepsilon$ additive error in $O(n^2 \log(1/\varepsilon))$ iterations. To make the statement more general, we note that ellipsoid method can be used for non-differentiable functions.

**Theorem 2.3.1.** *Let $x^{(k)}$ be the sequence of points produced by the cutting plane framework (Algorithm 3) for a convex function $f$. Given a function $\mathcal{V}$ mapping some subsets of $\mathbb{R}^n$, the set of subsets the algorithm uses, to non-negative number such that*

1. *(Linearity) For any set $E \subseteq \mathbb{R}^n$, any vector $v$ and any scalar $\alpha \geq 0$, we have $\mathcal{V}(\alpha E + v) = \alpha \mathcal{V}(E)$ where $\alpha E + v = \{\alpha x + v : x \in E\}$.*
2. *(Monotonic) For any set $F \subset E$, we have that $\mathcal{V}(F) \leq \mathcal{V}(E)$.*

*Then, for any set $\Omega \subset E^{(0)}$, we have that*

$$\min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{y\in\Omega} f(y) \leq \frac{\mathcal{V}(E^{(k)})}{\mathcal{V}(\Omega)} \cdot \left(\max_{z\in\Omega} f(z) - \min_{x\in\Omega} f(x)\right) .$$

*Remark* 2.3.2. We can think $\mathcal{V}(E)$ as some way to measure the size of $E$. It can be radius, mean-width or any other way to "size". For the ellipsoid method, we use $\mathcal{V}(E) = \mathrm{vol}(E)^{\frac{1}{n}}$ because we have proved the volume decreases in Lemma 2.2.1. We raise the volume to $\frac{1}{n}$-th power to satisfies the linearity.

*Proof.* Let $x^*$ be any minimizer of $f$ over $\Omega$. For any $\alpha > \frac{\mathcal{V}(E^{(k)})}{\mathcal{V}(\Omega)}$ and $S = (1-\alpha)x^* + \alpha\Omega$. By the linearity of $\mathcal{V}$, we have that

$$\mathcal{V}(S) = \alpha \mathcal{V}(\Omega) > \mathcal{V}(E^{(k)})$$

Therefore, $S$ is not a subset of $E^{(k)}$ and hence there is a point $y \in S \backslash E^{(k)}$. $y$ is not in $E^{(k)}$. This means it is separated by the gradient at some step $i \leq k$, namely

$$\nabla f(x^{(i)})^\top (y - x^{(i)}) > 0.$$

By the convexity of $f$, we have $f(x^{(i)}) \leq f(y)$. Since $y \in S$, we have $y = (1-\alpha)x^* + \alpha z$ for some $z \in \Omega$. Thus, the convexity of $f$ implies that

$$f(x^{(i)}) \leq f(y) \leq (1-\alpha) f(x^*) + \alpha f(z).$$

Therefore, we have

$$\min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{x\in\Omega} f(x) \leq \alpha \left(\max_{z\in\Omega} f(z) - \min_{x\in\Omega} f(x)\right).$$

Since this holds for any $\alpha > \frac{\mathcal{V}(E^{(k)})}{\mathcal{V}(\Omega)}$, we have the result. □

Combining Lemma 2.2.1 and Theorem 2.3.1, we have the following rate of convergence.

**Theorem 2.3.3.** *Let $f$ be a convex function on $\mathbb{R}^n$, $E^{(0)}$ be any initial ellipsoid and $\Omega \subset E^{(0)}$ be any convex set. Suppose that for any $x \in E^{(0)}$, we can find, in time $\mathcal{T}$, a nonzero vector $g(x)$ such that*

$$f(y) \geq f(x) \text{ for any } y \text{ such that } g(x)^\top (y - x) \geq 0.$$

*Then, we have*

$$\min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{y\in\Omega} f(y) \leq \left(\frac{\mathrm{vol}(E^{(0)})}{\mathrm{vol}(\Omega)}\right)^{\frac{1}{n}} \exp\left(-\frac{k}{2n(n+1)}\right) \left(\max_{z\in\Omega} f(z) - \min_{x\in\Omega} f(x)\right) .$$

*Furthermore, each iteration takes $O(n^2 + \mathcal{T})$ time.*

*Remark* 2.3.4. We usually call $g(x)$ the separation oracle.

*Proof.* Lemma 2.2.1 shows that the volume of the ellipsoid maintained decreases by $e^{-\frac{1}{2n+2}}$ every iteration. Hence, $\mathrm{vol}^{\frac{1}{n}}$ decreases by $\exp(-\frac{1}{2n(n+1)})$ every iteration. The bound follows by applying Theorem 2.3.1 with $\mathcal{V}(E) \overset{\mathrm{def}}{=} \mathrm{vol}(E)^{\frac{1}{n}}$. Next, we note the proof of Theorem 2.3.1 only used the fact that one side of halfspace defined by the gradient has higher value. Therefore, we can replace the gradient with the vector $g(x)$. $\square$

This theorem can be used to solve many problems in polynomial time. As an illustration, we show how to solve linear programs in polynomial time here.

**Theorem 2.3.5.** *Given a linear program $\min_{x\in\Omega} c^\top x$ where $\Omega = \{x \text{ such that } Ax \geq b\}$. Let the diameter $R \overset{\mathrm{def}}{=} \max_{x\in\Omega} \|x\|_2$ and the volume radius $r = \mathrm{vol}(\Omega)^{\frac{1}{n}}$. Then, we can find $x$ such that*

$$f(x) - \min_{x\in\Omega} f(x) \leq \varepsilon \cdot (\max_{x\in\Omega} c^\top x - \min_{x\in\Omega} c^\top x)$$

*in $O(n^2(n^2 + \mathrm{nnz}(A)) \log(\frac{R}{r\varepsilon}))$ time where $\mathrm{nnz}(A)$ is the number of non-zeros in $A$.*

*Remark* 2.3.6. If the dimension $n$ is constant, this algorithm is nearly linear time (linear to the number of constraints)!

*Proof.* For the linear program $\min_{Ax\geq b} c^\top x$, the function we want to minimize is

$$L(x) = c^\top x + \ell_{Ax\geq b} \quad \text{where} \quad \ell_{Ax\geq b}(x) = \begin{cases} 0 & \text{if } a_i^\top x \geq b_i \text{ for all } i \\ +\infty & \text{otherwises} \end{cases}. \tag{2.3.1}$$

For this function $L$, we can use the separation oracle $g(x) = c$ if $Ax \geq b$ and $g(x) = a_i$ if $a_i^\top x < b$.

We can simply pick $E^{(0)}$ be a unit ball centered at 0 with radius $R$ and apply Theorem 2.3.3 to find $x$ such that

$$f(x) - \min_{x\in\Omega} f(x) \leq \varepsilon(\max_{x\in\Omega} c^\top x - \min_{x\in\Omega} c^\top x)$$

in time $O(n^2(n^2 + \mathrm{nnz}(A)) \log(\frac{R}{r\varepsilon}))$. $\square$

To get the solution exactly, i.e., $\varepsilon = 0$, we need to assume the linear program is integral and the running time depends on the bound of the number in the matrix $A$ and the vectors $b$ and $c$. It is still open how to solve linear program in time polynomial only in the number of variables and constraints. We call such a running time *strongly polynomial.*

**Open Problem.** Can we solve linear programs in strongly polynomial time?

## 2.4 Center of Gravity Method

In the center of gravity method, we start with any bounded convex set containing a minimizer, e.g., a large enough cube, and the set maintained is simply the intersection of all halfspaces used so far.

The following classical theorem shows that the volume of the convex body decreases by at least a $1 - \frac{1}{e}$ factor. For computation purpose, it is important to establish a stable version of the theorem that does not require an exact center of gravity.

**Theorem 2.4.1** (Robust Grunbaum)**.** *Let $p$ be an isotropic logconcave distribution, namely $\mathbf{E}_{x\sim p} x = 0$ and $\mathbf{E}_{x\sim p} x^2 = 1$. For any $\theta \in \mathbb{R}^n$, $t \in \mathbb{R}$ we have*

$$\mathbf{P}_{x\sim p}(x^\top\theta \geq t) \geq \frac{1}{e} - t.$$

*Proof.* By taking the marginal with respect to the direction $\theta$, we can assume the distribution is one dimensional. Let $P(t) = \mathbf{P}_{x\sim p}(x^\top\theta \le t)$. Note that $P(t)$ is the convolution of $p$ and $1_{(-\infty,0]}$. Hence, it is logconcave (Lemma 1.4.2). By some limiting arguments, we can assume $P(-M) = 0$ and $P(M) = 1$. Since $\mathbf{E}_{x\sim p}x = 0$, we have that

$$\int_{-M}^{M} tP'(t) = 0$$

Integration by parts gives that $\int_{-M}^{M} P(t)dt = M$. Note that $P(t)$ is increasing logconcave, if $P(0)$ is too small, it would make $\int_{-M}^{M} P(t)dt$ too small. To be precise, since $P$ is logconcave, we have that

$$-\log P(t) \ge -\log P(0) - \frac{P'(0)}{P(0)}t.$$

Or we simply write $P(t) \le P(0)e^{\alpha t}$ for some $\alpha$. Hence,

$$M = \int_{-M}^{M} P(t)dt \le \int_{-\infty}^{\frac{1}{\alpha}} P(0)e^{\alpha t}dt + \int_{1/\alpha}^{M} 1dt = \frac{eP(0)}{\alpha} + M - \frac{1}{\alpha}.$$

This shows that $P(0) \ge \frac{1}{e}$.

Next, Lemma 2.4.2 shows that $\max_x p(x) \le 1$. Therefore, the cumulative distribution $P$ is 1-Lipschitz and we

$$\mathbf{P}_{x\sim p}(x^\top\theta \ge t) \ge \mathbf{P}_{x\sim p}(x^\top\theta \ge 0) - t \ge \frac{1}{e} - t.$$

□

**Lemma 2.4.2.** *Let $p$ be a one-dimensional isotropic logconcave density. Then* $\max p(x) \le 1$.

For a proof of this (and for other properties of logconcave functions), we refer the reader to [45].

**Exercise 2.4.3.** Give a short proof that $\max_x p(x) = O(1)$ for any one-dimensional isotropic logconcave density.

Using the robust Grunbaum theorem 2.4.1, we get the following algorithm.

**Lemma 2.4.4.** *Consider the randomized center of gravity method defined as follows:*

$$x^{(k)} = \frac{1}{N}\sum_{i=1}^{N} y^i,$$
$$E^{(k+1)} = E^{(k)} \cap H^{(k)}$$

*where $y^i$ are i.i.d. uniform random points from $E^{(k)}$. Then*

$$\mathbf{E}(\text{vol}(E^{(k+1)})) \le \left(1 - \frac{1}{e} + \sqrt{\frac{n}{N}}\right) \text{vol}(E^{(k)}).$$

*Proof.* Without loss of generality, we assume that $E^{(k)}$ is in isotropic position, i.e., $\mathbf{E}(y^i) = 0$ and $\mathbf{E}(y^i(y^i)^\top) = I$. Then we have $\mathbf{E}(x^{(k)}) = 0$ and

$$\mathbf{E}\left\|x^{(k)}\right\|^2 = \frac{1}{N}\mathbf{E}\left\|y^i\right\|^2 = \frac{n}{N}.$$

Therefore,

$$\mathbf{E}\left\|x^{(k)}\right\| \le \sqrt{\mathbf{E}\left(\left\|x^{(k)}\right\|^2\right)} = \sqrt{\frac{n}{N}}.$$

Thus, we can apply Theorem 2.4.1 with $t = \sqrt{\frac{n}{N}}$. □

Theorem 2.3.1 readily gives the following guarantee for convex optimization, again using volume radius as the measure of progress.

**Theorem 2.4.5.** *Let $f$ be a convex function on $\mathbb{R}^n$, $E^{(0)}$ be any initial set and $\Omega \subset E^{(0)}$ be any convex set. Suppose that for any $x \in E^{(0)}$, we can find a nonzero vector $g(x)$ such that*

$$f(y) \geq f(x) \text{ for any } y \text{ such that } g(x)^\top (y - x) \geq 0.$$

*Then, for the center of gravity method with $N = 10n$, we have*

$$\mathbf{E} \min_{i=1,2,\cdots k} f(x^{(i)}) - \min_{y \in \Omega} f(y) \leq \left(\frac{\mathrm{vol}(E^{(0)})}{\mathrm{vol}(\Omega)}\right)^{\frac{1}{n}} (0.95)^{\frac{k}{n}} \left(\max_{z\in\Omega} f(z) - \min_{x \in \Omega} f(x)\right) .$$

## Discussion

In later chapters we will see how to implement each iteration of the center of gravity method in polynomial time. Computing the exact center of gravity is #P-hard even for a polytope, but we can find an arbitrarily close approximation in randomized polynomial time by sampling. The method generalizes to certain noisy computation models.

We will also see that the iteration complexity of the center of gravity method, $\tilde{O}(n)$ iterations, is asymptotically optimal. The complexity of each iteration, and the complexity of convex optimization as well as the special case of linear programming continues to be an active research area.

Getting a cutting plane algorithm that is fast in both theory and practice is still an active research area. As an illustration, consider the following open problem.

**Open Problem.** Suppose we take $E^{(k+1)} = H^{(k)} \cap E^{(k)}$ and measure the progress by the volume of $E^{(k)}$. One natural idea is to select a point that decreases the volume as much as possible for the next iteration. In discrete geometry, [7] showed that this point can be approximated in polynomial time via sampling. As far as I know, this has not been used in convex optimization. So, is this greedy method for convex optimization practical? More radically, suppose we planned to run the algorithm for only $T$ iterations, then can we find a point that decreases the volume at the $T^{th}$ iteration most among all future responses from the gradient?

## 2.5 Cutting Plane method for Volume Computation

The cutting plane method gives a simple algorithm for computing the volume of a bounded set, or the integral of a function. The algorithm relies crucially on the ability to compute the center of gravity of the original set/function restricted by halfspaces. For simplicity we describe it here for the case when the input object is a convex body.

**Algorithm 5:** `CuttingPlaneVolume`

```
Input: A body $K \subset \mathbb{R}^n$, $r \in R$, and an oracle for computing centroid.
Fix an ordering on the axes: $e_1, e_2, \ldots e_n$
$K^{(0)} = K$, $z^{(0)} = \mathrm{centroid}(K^{(0)}), V = 1, k = 0$.
for $k = 0, \cdots$ do
    Let $e_i$ be an axis vector so that the width of $K^{(k)}$ along $e_i$ is greater than $r$.
    if There is no such $e_i$ then Break.
    Let $a$ be $e_i$ or $-e_i$ with sign chosen so that $H^{(k)} \overset{\mathrm{def}}{=} \left\{x : a^\top x \leq a^\top z^{(k)}\right\}$ contains $z^{(0)}$.
    $K^{(k+1)} \leftarrow K^{(k)} \cap H^{(k)}$.
    $z^{(k+1)} \leftarrow \mathrm{centroid}(K^{(k+1)})$.
    $\widehat{z} \leftarrow \mathrm{centroid}(K^{(k)} \setminus K^{(k+1)})$.
    $V \leftarrow V \cdot \frac{\|\widehat{z} - z^{(k+1)}\|}{\|\widehat{z} - z^{(k)}\|}$.
end
Return $V \cdot \prod_{i=1}^n w_i(K^{(k+1)})$ where $w_i(\cdot)$ is the width along $e_i$.
```

The idea behind the algorithm is simple: when we cut a set with a hyperplane through its centroid, the line joining the centroids of the two sides passes through the original centroid; moreover, the ratio of the two segments is exactly the ratio of the volumes of the two halfspaces.

**Lemma 2.5.1.** *For any measurable bounded set $S$ in $\mathbb{R}^n$ and any halfspace $H$ with bounding hyperplane containing the centroid of $S$, we have*

$$centroid(S) = \frac{\mathrm{vol}(S \cap H)}{\mathrm{vol}(S)} centroid(S \cap H) + \frac{vol(S \cap \overline{H})}{\mathrm{vol}(S)} centroid(S \cap \overline{H}).$$

The following lemma has a proof similar to that of the Grunbaum theorem.

**Lemma 2.5.2.** *Let $K \subseteq \mathbb{R}^n$ be a convex body with centroid at the origin. Suppose that for some unit vector $\theta$, the support of $K$ along $\theta$ is $[a, b]$. Then,*

$$|a| \geq \frac{b}{n}.$$

**Exercise 2.5.3.** Prove Lemma 2.5.2. [Hint: Use Theorem 1.4.4.]

Using the above property, we can show that the algorithm reaches a cuboid in a small number of iterations.

**Theorem 2.5.4.** *Let $K$ be a convex body in $\mathbb{R}^n$ containing a cube of side length $r$ around its centroid and contained in a cube of side length $R$. Algorithm* `CuttingPlaneVolume` *correctly computes the volume of $K$ using $O(n \log \frac{nR}{r})$ centroid computations.*

*Proof.* By Lemma 2.4.1, at each iteration, the volume of the remaining set $K^{(k)}$ decreases by a factor of at most $(1 - \frac{1}{e})$. When the directional width along any axis is less than $r/2$ (namely $\max_{x \in K} \left|e_i^\top x\right| \leq \frac{r}{2}$), the algorithm stops cutting along that axis. Thus, by Lemma 2.5.2, the directional width along every axis is at least $r/2(n+1)$. Since the set always contains the origin, and the original set contains a cube of side length $r$, when the algorithm stops, it must be an axis-parallel cuboid. So the final volume is at least $(r/2n)^n$. The initial volume is at most $R^n$. Therefore the number of iterations is at most

$$\log_{(1-\frac{1}{e})} \left( \frac{R^n}{(r/2(n+1))^n} \right) = O\left(n \log(nR/r)\right).$$

In each iteration, by Lemma 2.5.1, the algorithm maintains the ratio of the volume of the original $K$ to the current $K^{(k)}$. □

The above algorithm shows that computing the volume is polytime reducible to computing the centroid. Since volume is known to be #P-hard for explicit polytopes, this means that centroid computation is also #P-hard for polytopes [52]. In later chapters we will see randomized polytime algorithms for sampling and hence for approximating centroid and volume.

## 2.6 Sphere and Parabola Method

In Section 1.5, we proved that gradient descent converges at the rate $(1 - \frac{\mu}{L})^k$ in function value assuming the function satisfies $\mu \cdot I \preceq \nabla^2 f(x) \preceq L \cdot I$ for all $x$. Hence, if $\frac{L}{\mu}$ is not very large, this rate on the function value can be much better than the rate $\left(1 - \frac{1}{n^2}\right)^k$ of ellipsoid method or the rate $\left(1 - \frac{1}{n}\right)^k$ of the center of gravity method (recall that the convergence rate in function value is $n$ times slower than the convergence rate in volume.) In this section, we show how to modify ellipsoid method to get a faster convergence rate when $\frac{L}{\mu}$ is small. One can view this whole section as just a reinterpretation of accelerated gradient descent (which we haven't seen yet) in the cutting plane framework. In a later section, we will give another interpretation.

## Sphere Method

For a strongly convex function, the optimal $x^*$ is contained in a strictly smaller region than a halfspace. In particular, we have

$$f(x^*) \geq f(x) + \nabla f(x)^\top (x^* - x) + \frac{\mu}{2} \|x^* - x\|_2^2 .$$

Completing the squares, we have that

$$\left\|x^* - x^{++}\right\|_2^2 \leq \frac{\|\nabla f(x)\|_2^2}{\mu^2} - \frac{2}{\mu}(f(x) - f(x^*)) \tag{2.6.1}$$

where $x^{++} \overset{\text{def}}{=} x - \frac{1}{\mu}\nabla f(x)$. To use this formula in the cutting plane framework, we need a crude upper bound on $f(x^*)$. Certainly, one can simply use $f(x^*) \leq f(x)$. Or, we can use the assumption on $L$ (Lemma 1.5.3) and get

$$f(x^*) \leq f(x - \frac{1}{L}\nabla f(x)) \leq f(x) - \frac{1}{2L}\|\nabla f(x)\|^2.$$

Putting it into (2.6.1), we know that $x^*$ lies in a sphere centered at $x^{++}$ with squared radius at most

$$(1 - \frac{\mu}{L}) \cdot \frac{\|\nabla f(x)\|_2^2}{\mu^2}. \tag{2.6.2}$$

Another issue of ellipsoid method is the $O(n^2)$ cost per each iteration. To make each iteration cheaper, we will a ball instead of an ellipsoid. We have the following algorithm.

**Algorithm 6:** `SphereMethod`

```
Input: Initial point $x^{(0)} \in \mathbb{R}^n$, the strong convexity parameter $\mu$, and the Lipschitz gradient parameter $L$.
$Q^{(0)} \leftarrow \mathbb{R}^n$.
for $k = 0, \cdots$ do
    Set $Q = \left\{x \in \mathbb{R}^n : \left\|x - (x^{(k)} - \frac{1}{\mu}\nabla f(x^{(k)}))\right\|_2^2 \leq \frac{1-\frac{\mu}{L}}{\mu^2} \cdot \left\|\nabla f(x^{(k)})\right\|_2^2\right\}$.
    $Q^{(k+1)} \leftarrow$ minSphere($Q \cap Q^{(k)}$) where minSphere($K$) is the smallest sphere covering $K$.
    Set $x^{(k)}$ be the center of $Q^{(k)}$
end
```

To analyze `SphereMethod`, we need the following lemma, which is illustrated in Figure 2.6.1.

**Lemma 2.6.1.** *For any $g \in \mathbb{R}^n$ and $\epsilon \in (0,1)$, we have $B(0,1) \cap B(g, \|g\|_2\sqrt{1-\epsilon}) \subset B(x, \sqrt{1-\epsilon})$ for some $x$.*

*Proof.* It suffices to consder the two-dimensional case by symmetry, and to assume that $g = ae_1$. Let $(x, 0)$ be the center of the smallest ball containing the required intersection, and $y$ be its radius. Then $x^2 + y^2 = 1$ and $(x-a)^2 + y^2 = (1-\epsilon)a^2$. This implies that

$$x = \frac{1+\epsilon a^2}{2a}$$

and so

$$y^2 = 1 - \frac{1}{4a^2} - \frac{\epsilon}{2} - \frac{\epsilon^2 a^2}{4} \leq 1 - \epsilon$$

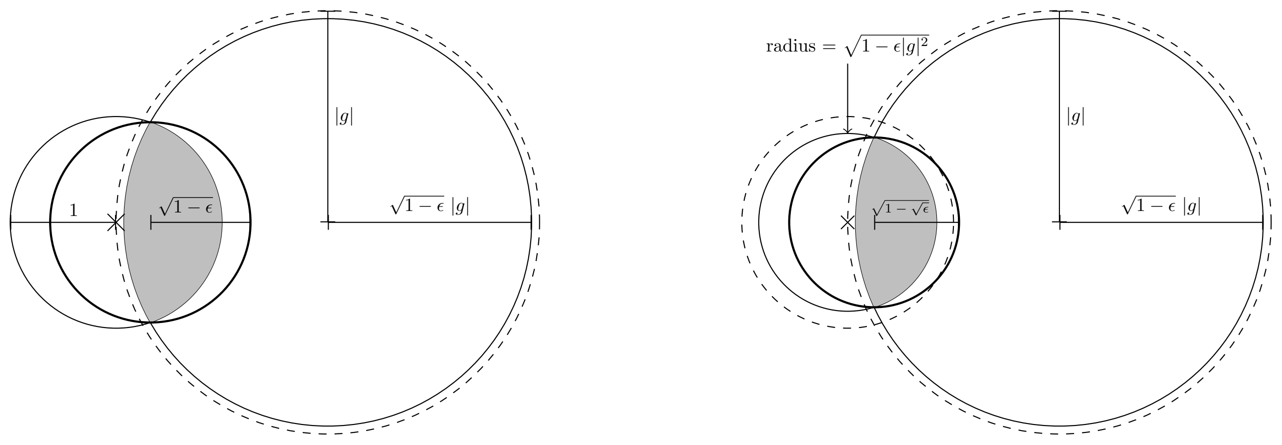
as claimed. $\square$

**Exercise 2.6.2.** Prove the following extension of Lemma 2.6.1: There exists $x$ s.t. $B(0, \sqrt{1-\epsilon|g|^2}) \cap B(g, |g|\sqrt{1-\epsilon}) \subset B(x, \sqrt{1-\sqrt{\epsilon}})$.

Since we will give an improved algorithm next based on the above exercise, we only give a draft of the proof here.

**Lemma 2.6.3.** *Let the measure of progress $\mathcal{V}(Q) = \mathrm{radius}(Q)$. Then, we have that $x^* \in Q^{(k)}$ and $\mathcal{V}(Q^{(k+1)}) \leq \sqrt{1 - \frac{\mu}{L}} \cdot \mathcal{V}(Q^{(k)})$ for all $k$.*

*Remark* 2.6.4. The function value decrease then follows from Theorem 2.3.1.



$B(0,1) \cap B(g, |g|\sqrt{1-\epsilon}) \subset B(x, \sqrt{1-\epsilon})$ $B(0, \sqrt{1-\epsilon|g|^2}) \cap B(g, |g|\sqrt{1-\epsilon}) \subset B(x, \sqrt{1-\sqrt{\epsilon}})$

**Figure 2.6.1:** The left diagram shows the intersection shrinks at the same rate if only one of the ball shrinks; the right diagram shows the intersection shrinks much faster if two balls shrinks at the same absolute amount.

*Proof Sketch.* The fact $x^* \in Q^{(k)}$ follows directly from the definition of $Q$. For the decrease of radius, suppose that $Q^{(k)} = \{x \in \mathbb{R}^n : \|x - x^{(k)}\| \leq R^{(k)}\}$. Then, the new sphere is given by

$$Q^{(k+1)} = \texttt{minSphere}\left(\left\{\left\|x - (x^{(k)} - \frac{1}{\mu}\nabla f(x^{(k)}))\right\|_2^2 \leq \frac{1-\frac{\mu}{L}}{\mu^2} \cdot \left\|\nabla f(x^{(k)})\right\|_2^2\right\} \cap \left\{\|x - x^{(k)}\| \leq R^{(k)}\right\}\right).$$

To compute $\frac{\text{radius}(Q^{(k+1)})^2}{\text{radius}(Q^{(k)})^2}$, we can assume $x^{(k)} = 0$ and $R^{(k)} = 1$ and let $g = \frac{\nabla f(0)}{\mu}$. Hence, we have

$$Q^{(k+1)} = \texttt{minSphere}\left(\left\{\|x - g\|_2^2 \leq (1 - \frac{\mu}{L}) \cdot \|g\|_2^2\right\} \cap \{\|x\| \leq 1\}\right).$$

Now Lemma 2.6.1 (with $\epsilon = \frac{\mu}{L}$) shows that the radius$(Q^{(k+1)})^2 \leq 1 - \frac{\mu}{L}$. □

Note that this gives exactly the same convergence rate as the gradient descent.

**Exercise 2.6.5.** Prove Lemma 2.6.1.

## Parabola Method Via Epigraph

In previous sections, we discussed cutting plane methods that maintain a region that contains the minimizer $x^*$ of $f$. The proof of such cutting plane method relies on the following inequality

$$f(x) > f(x^*) \geq f(x) + \langle \nabla f(x), x^* - x \rangle . \tag{2.6.3}$$

Notice that the left-hand side is a strict inequality (unless we already solved the problem). As the algorithm runs, we will find a new point $x^{(\text{new})}$ such that $f(x^{(\text{new})}) < f(x)$. Therefore, we should move the halfspaces we got before. We expect merely updating the halfspaces can improves the convergence rate from $1 - \frac{\mu}{L}$ to $1 - \sqrt{\frac{\mu}{L}}$ because of the right diagram in Figure 2.6.1. An efficient way to manage all this information is to directly maintain a region that contains $(x^*, f(x^*))$. Now, we can think the inequality $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$ as a cutting plane of the epigraph of $f$ and we do not need to update previous cutting planes anymore.

The new algorithm is an epigraph cutting plane method.

**Algorithm 7:** `ParabolaMethod`

```
Input: Initial point $x^{(0)} \in \mathbb{R}^n$ and the strong convexity parameter $\mu$.
$q^{(0)}(y) \leftarrow -\infty$.
for $k = 0, \cdots$ do
    Set $q^{(k+\frac{1}{2})}(y) = f(x^{(k)}) + \nabla f(x^{(k)})^\top (y - x^{(k)}) + \frac{\mu}{2}\left\|y - x^{(k)}\right\|^2$.
    Let $q^{(k+1)} = \texttt{maxParabola}(\max(q^{(k+\frac{1}{2})}, q^{(k)}))$ where $\texttt{maxParabola}(q)$ outputs the parabolic function $p$ such
      that $p(x) \leq q(x)$ for all $x$ with the maximum $\min_x p(x)$.
    // Alternatively, one can use $x^{(k+\frac{1}{2})} \leftarrow x^{(k)} - \frac{1}{L}\nabla f(x^{(k)})$.
    Let $x^{(k+\frac{1}{2})} = \texttt{lineSearch}(x^{(k)}, \nabla f(x^{(k)}))$ where
        $\texttt{lineSearch}(x, y) = \mathrm{argmin}_{z = x + t(x-y) \text{ with } t \in \mathbb{R}} f(z)$.
    Let $x^{(k+1)} = \texttt{lineSearch}(c^{(k+1)}, x^{(k+\frac{1}{2})})$ where $c^{(k+1)}$ be the minimizer of $q^{(k+1)}(y)$.
end
```

As in gradient descent, to avoid using the parameter $L$, it does a line search from $x^{(k)}$ along the direction $\nabla f(x^{(k)})$. The differences between `SphereMethod` and `ParabolaMethod` are the following:

1. It is an epigraph cutting plane method and does not need to know $f^*$.
2. Both ellipsoid method and `SphereMethod` evaluate $\nabla f$ at the center $c^{(k)}$ of the ellipsoid/sphere. However, the function value $f(c^{(k)})$ may be larger than some point we evaluated before. On the other hand, evaluating $\nabla f$ at $x^{(k+\frac{1}{2})}$ does not work either because the gradient at this point cannot cut the domain by "half" like the center of ellipsoid. Therefore, the algorithm instead uses a line search between the center $c^{(k)}$ and the previous gradient descent point $x^{(k+\frac{1}{2})}$. This makes sure we always evaluate $\nabla f$ at the point with smallest function value we have ever seen. This make sure that all the balls shrink every step. Furthermore, since this is a line search from the center, the cutting plane induced by the gradient must cut away the center. See the right diagram in Figure 2.6.1.

These are natural modifications to make the algorithm use available information. For the explicit formula in `maxParabola`, see [16] for the proof.

**Algorithm 8:** $q = \texttt{maxParabola}(\max(q_a, q_b))$

```
Input: $q_a(x) = v_A + \frac{\mu}{2}\|x - c_A\|_2^2$ and $q_B(x) = v_B + \frac{\mu}{2}\|x - c_B\|_2^2$.
Compute $\lambda = \mathrm{proj}_{[0,1]}\left(\frac{1}{2} + \frac{v_A - v_B}{\mu\|c_A - c_B\|^2}\right)$.
$c_\lambda = \lambda c_A + (1-\lambda) c_B$.
$v_\lambda = \lambda v_A + (1-\lambda) v_B + \frac{\mu}{2}\lambda(1-\lambda)\|c_A - c_B\|^2$.
Output: $q(x) = v_\lambda + \frac{\mu}{2}\|x - c_\lambda\|_2^2$.
```

The key fact we will be using in the formula is that

$$v_\lambda = v_A + \frac{\mu}{8}\frac{(\|c_A - c_B\|^2 + \frac{2}{\mu}(v_B - v_A))^2}{\|c_A - c_B\|^2}.$$

Namely, the quadratic lower bound improves a lot whenever $\frac{\mu}{2}\|c_A - c_B\|^2 \gg v_B - v_A$ or $\frac{\mu}{2}\|c_A - c_B\|^2 \ll v_B - v_A$. Using this, we can prove the `ParabolaMethod`.

**Theorem 2.6.6.** *Assume that $f$ is $\mu$-strongly convex with $L$-Lipschitz gradient. Let $r_k = f(x^{(k)}) - \min_y q^{(k)}(y)$. Then, we have that*

$$r_{k+1}^2 \leq (1 - \sqrt{\frac{\mu}{L}}) \cdot r_k^2.$$

*In particular, we have that*

$$f(x^{(k+1)}) - f^* \leq \frac{1}{2\mu}(1 - \sqrt{\frac{\mu}{L}})^k \|\nabla f(x^{(0)})\|^2.$$

*Remark* 2.6.7. Note that the radius square of $\{y : q^{(k)}(y) = f(x^{(k)})\}$ is $\frac{2}{\mu}(f(x^{(k)}) - \min_y q^{(k)}(y))$ because $q^{(k)}(y) = \min_y q^{(k)}(y) + \frac{\mu}{2}\|y - \arg\min_x q^{(k)}(x)\|^2$.

Hence, $r_k$ is measuring the "radius" of our quadratic lower bound. To relate to the cutting plane framework, we can think the set is $\{y : q^{(k)}(y) \leq f(x^{(k)})\}$ and the measure $\mathcal{V} = f(x^{(k)}) - q^{(k)}(y)$.

*Proof.* Fix some $k$. We write $q^{(k)}(y) = v_A + \frac{\mu}{2}\left\|y - c_A\right\|^2$ and $q^{(k+\frac{1}{2})}(y) = v_B + \frac{\mu}{2}\left\|y - c_B\right\|^2$ with

$$c_A = c^{(k)}, \quad c_B = x^{(k)} - \frac{\nabla f(x^{(k)})}{\mu} \quad \text{and} \quad v_B = f(x^{(k)}) - \frac{\|\nabla f(x^{(k)})\|^2}{2\mu}.$$

Using the notation in `maxParabola`, we write $q^{(k+1)}(y) = v_\lambda + \frac{\mu}{2}\left\|y - c_\lambda\right\|^2$. Note that $r_{k+1}^2 = f(x^{(k+1)}) - v_\lambda$ and $r_k^2 = f(x^{(k)}) - v_A$. Therefore, we have

$$\frac{r_k^2 - r_{k+1}^2}{r_k^2} = \frac{f(x^{(k)}) - f(x^{(k+1)}) + v_\lambda - v_A}{r_k^2}. \tag{2.6.4}$$

To bound the right hand side, it suffices bound $v_A$ and $v_\lambda$. From the description of the algorithm `maxParabola`, we see that there are three cases $\lambda = 0$, $\lambda = 1$ and $0 < \lambda < 1$. We only focus on proving the nontrivial case $\lambda \in (0, 1)$. In this case, we have that

$$\begin{aligned} v_\lambda &= v_A + \frac{\mu}{8}\frac{(\|c_A - c_B\|^2 + \frac{2}{\mu}(v_B - v_A))^2}{\|c_A - c_B\|^2} \\ &= v_A + \frac{\mu}{8}\frac{(\|c_A - c_B\|^2 + \frac{2}{\mu}(f(x^{(k)}) - v_A) - \frac{\|\nabla f(x^{(k)})\|^2}{\mu^2})^2}{\|c_A - c_B\|^2}. \end{aligned}$$

Since $x^{(k)}$ is a line search from $c^{(k)}$ and that $f \geq q^{(k)}$, we have $f(x^{(k)}) \geq f(c^{(k)}) \geq q^{(k)}(c^{(k)}) \geq v_A$. Next, we claim that $\|c_A - c_B\|^2 \geq \frac{\|\nabla f(x^{(k)})\|^2}{\mu^2}$. Using these two facts, we can prove that

$$v_\lambda \geq v_A + \frac{\mu}{8}\frac{(\frac{2}{\mu}(f(x^{(k)}) - v_A))^2}{\frac{\|\nabla f(x^{(k)})\|^2}{\mu^2}} = v_A + \frac{\mu \cdot r_k^4}{2\|\nabla f(x^{(k)})\|^2}.$$

Putting this into (2.6.4), we have

$$\begin{aligned} \frac{r_k^2 - r_{k+1}^2}{r_k^2} &\geq \frac{f(x^{(k)}) - f(x^{(k+1)})}{r_k^2} + \frac{\mu \cdot r_k^2}{2\|\nabla f(x^{(k)})\|^2} \\ &\geq \frac{f(x^{(k+\frac{1}{2})}) - f(x^{(k+1)})}{r_k^2} + \frac{\mu \cdot r_k^2}{2\|\nabla f(x^{(k)})\|^2} \\ &\geq \frac{\|\nabla f(x^{(k)})\|^2}{2Lr_k^2} + \frac{\mu \cdot r_k^2}{2\|\nabla f(x^{(k)})\|^2} \\ &\geq \sqrt{\frac{\mu}{L}} \end{aligned} \tag{2.6.5}$$

where the second inequality is due to the fact $x^{(k)}$ is a line search from $x^{(k+\frac{1}{2})}$, the third inequality is due to the assumption on $L$ (Lemma 1.5.3), the last inequality follows from Cauchy-Schwarz inequality.

For the final conclusion, we note that

$$\begin{aligned} q^{(1)}(y) &= f(x^{(0)}) + \nabla f(x^{(0)})^\top (y - x^{(0)}) + \frac{\mu}{2}\left\|y - x^{(0)}\right\|^2 \\ &= f(x^{(0)}) - \frac{1}{2\mu}\|\nabla f(x^{(0)})\|^2 + \frac{\mu}{2}\left\|y - (x^{(0)} - \frac{1}{\mu}\nabla f(x^{(0)}))\right\|^2. \end{aligned}$$

Hence, we have $v^{(1)} = f(x^{(0)}) - \frac{1}{2\mu}\|\nabla f(x^{(0)})\|^2$ and hence $r_1 \leq f(x^{(1)}) - v^{(1)} \leq \frac{1}{2\mu}\|\nabla f(x^{(0)})\|^2$.

Finally, to prove the claim, we note that $x^{(k)}$ is the result of the linear search of $f$ between $c^{(k)}$ and some point. Therefore, we have that $\nabla f(x^{(k)}) \perp (x^{(k)} - c^{(k)})$ and hence

$$\|c_A - c_B\|^2 = \|c^{(k)} - x^{(k)} + \frac{\nabla f(x^{(k)})}{\mu}\|^2 \geq \frac{\|\nabla f(x^{(k)})\|^2}{\mu^2}.$$

□

*Remark* 2.6.8. We can view (2.6.5) as the key equation of the proof above. It shows the progress is roughly $\frac{\|\nabla f\|^2}{L} + \frac{\mu}{\|\nabla f\|^2}$ where the first term comes from the progress we get on the function value and the second term comes from "the curvature" of cutting sphere.

### Discussion

This section is about the idea of managing cutting planes, and as a byproduct we get an accelerated rate of convergence. As we will see later, standard accelerated gradient descent does not use line search and achieves the rate $1-\sqrt{\frac{\mu}{L}}$. However, it seems that the use of line search helps in practice and that with a careful implementation, line search can be as cheap as gradient computation. For more difficult problems, one may want to store multiple quadratic lower bounds (see [16]).

## 2.7 Lower Bounds

In this chapter, we have discussed few cutting plane methods. In particular, we showed that $O(\min(n,\sqrt{\frac{L}{\mu}})\log(\frac{1}{\epsilon}))$ many gradient computations suffice. We conclude this section by showing that $\min(n,\sqrt{\frac{L}{\mu}})$ is in fact optimal among "gradient-based" methods.

**Theorem 2.7.1.** *Consider the function*

$$f(x) = L\cdot\left(-x_1+\frac{1}{2}x_1^2+\frac{1}{2}\sum_{k=1}^{n-1}(x_k-x_{k+1})^2+\frac{1}{2}x_n^2\right)+\frac{\mu}{2}\sum_{k=0}^{n}x_k^2. \tag{2.7.1}$$

*Assume that our algorithm satisfies* $x^{(k)} \in \operatorname{span}(x^{(0)},\nabla f(x^{(0)}),\nabla f(x^{(1)}),\cdots,\nabla f(x^{(k-1)}))$ *with the initial point* $x^{(0)}=0$. *Then, we have that*

$$f(x_k)-f^* \geq \left(1-\Omega\left(\sqrt{\frac{\mu}{L}}\right)\right)^k \frac{\mu}{2}\|x^{(0)}-x^*\|^2$$

*for* $k<n$.

*Proof draft.* Note that the gradient at $x^{(0)}$ is of the form $(?,0,0,0,\cdots)$ and hence by the assumption $x^{(1)} = (?,0,0,0,\cdots)$ and $\nabla f(x^{(1)}) = (?,?,0,0,\cdots)$. By induction, only the first $k$ coordinates of $x^{(k)}$ are non-zero. However, one can check that the minimizer of the function (2.7.1) has roughly $\sqrt{\frac{L}{\mu}}$ non-zeros. Hence, it takes $\Omega(\sqrt{\frac{L}{\mu}})$ steps to get an approximate solution. For a complete proof, see Section 2.1.4 in [48]. □

Now we give an explanation where the term $\sqrt{\frac{L}{\mu}}$ comes from. Omitting some terms and approximating the summation by an integral, we can view the problem as

$$f(x)=\frac{L}{2}\int_0^\infty x'(t)^2dt+\frac{\mu}{2}\int_0^\infty x(t)^2dt$$

with the boundary condition $x(0)=1$ and $x(\infty)=0$. Computing the directional derivative of $f$, we have

$$\begin{aligned} Df(x)[h] &= L\int_0^\infty x'(t)h'(t)dt+\mu\int_0^\infty x(t)h(t)dt \\ &= -L\int_0^\infty x''(t)h(t)dt+\mu\int_0^\infty x(t)h(t)dt. \end{aligned}$$

Therefore, we can write $\nabla f(x) = -Lx'' + \mu x$. Hence, the optimal solution is given by the ODE $x'' = \frac{\mu}{L}x$. Solving this ODE with $x(0) = 1$ and $x(\infty) = 0$, we get

$$x(t) = e^{-\sqrt{\frac{\mu}{L}}t}.$$

Therefore, the solution vanishes after $t \gg \sqrt{\frac{L}{\mu}}$. Note that in the original problem, the bound breaks when the algorithm reaches all variables which takes $n$ iterations. This is the reason why this gives a bound of $\Theta(\min(\sqrt{\kappa}, n))$.

Note that this "worst" function naturally appears in many problems. So, it is a problem we need to address. In some sense, the proof points out a common issue of any algorithm which only uses gradient information. Given any convex function, we construct the dependence graph $G$ on the set of variables $x_i$ by connecting $x_i$ to $x_j$ if $\nabla f(x)_i$ depends on $x_j$ or $\nabla f(x)_j$ depends on $x_i$ (given all other variables). Note the the dependence graph $G$ of the worst function is simply a $n$ vertex path, whose diameter is $n-1$. Also, note that gradient descent can only transmit information from one vertex to another in each iteration. Therefore, it takes at least $\Omega(\text{diameter})$ time to solve the problem unless if we know the solution is sparse (when $L/\mu$ is small). However, we note that this is not a lower bound for all algorithms.

The problem (2.7.1) belongs to a general class of functions called Laplacian systems and it can be solved in nearly linear time using spectral graph theory. Recently, nearly linear time algorithms have been developed for various Laplacian systems.

Chapter 3

# Reduction

In the previous two chapters, we saw various algorithms to minimize a convex function.They have different runtimes and seems difficult to compare. Before going deeper and discussing more results, in this chapter, we will focus on how to understand these different results and their relationships, via reductions between them. We will do this at three different levels. The most abstract is the level of oracles, which we will define presently; the second is black-box algorithms which access the function in restricted ways, e.g., via the gradient; and the last is with full information and under additional structural assumptions.

## 3.1 State of the Art

Here we summarize the state-of-the-art. We tried to keep the table short, and there are many important results omitted.

## 3.2 Remarks on the Runtime

### Structured vs black-box problems

Table 3.1 shows the runtime for some convex problems assuming some restricted way to access the input convex function. We call the way to access the function an *oracle*. For example, we say we only have a first-order oracle if we can only access the function using the gradient of the function. In this setting, we are particularly interested in how many times we need to use the oracle, such as how many gradients we need to compute. Since the function is only accessed through an oracle, we can usually bound how many calls are needed to "learn" the function. In fact, all algorithms in the table are optimal in terms of the number of oracle calls. This is in huge contrast to the structured problems where all of the results are open.

Table 3.2 has the runtimes for some general convex problems. Unlike black-box problems, these algorithms assume the full description of the problem. For this setting, we usually aim at getting a runtime that depends on the structure of the problem instead of the numerical values of the input. Usually, such algorithms converge with a rate of $\log(1/\epsilon)$. For all of these problem, the best possible runtime is open. Finally, we note that the last two problems, linear programs and empirical risk minimization are complete in the sense that they can be used to approximate any convex function. Therefore, the last results seem to suggest that general convex problems are as easy as general linear systems for now.

### Understanding the bounds

All the runtimes should have the correct "units". For instance, if the unit of the function value is $J$ (Joule) and the unit of the space is $m$, then the units of $G$, $R$ and $\epsilon$ are $Jm^{-1}$, $m$ and $J$ respectively. Therefore, the number of iterations of gradient descent $\frac{GR}{\epsilon}$ is unitless. In the table, we ignored some entries inside the log which leads to an "incorrect" unit. However, when everything is written correctly, the number of iterations must be unitless. Furthermore, we can view $GR$, $LR^2$, $\frac{G^2}{\mu}$ all as upper bounds of the initial error $f(x^{(0)}) - f^*$. For example, we know that

$$f(x^{(0)}) - f^* \leq \left\langle \nabla f(x^{(0)}), x^{(0)} - x^* \right\rangle \leq GR$$

and that

$$f(x^{(0)}) - f^* \leq \left\langle \nabla f(x^*), x^{(0)} - x^* \right\rangle + \frac{L}{2}\|x^{(0)} - x^*\|^2 \leq LR^2.$$

Therefore, the terms $\frac{GR}{k}$, $\frac{G^2}{\mu k}$ and $\frac{LR^2}{k}$ are all just measuring how much we need to decrease multiplicatively from the initial error. With this view, we can think the number of iteration required for Lipschitz convex functions, strongly convex functions and smooth functions are exactly $\frac{1}{\epsilon^2_{\text{relative}}}$, $\frac{1}{\epsilon_{\text{relative}}}$ and $\frac{1}{\sqrt{\epsilon_{\text{relative}}}}$.

### $L/\mu$ can be dimension dependent

The ratio $L/\mu$ heavily depends on the application. For some easy problems, it can be as small as $O(1)$. However, it can be very large for many natural problems. Here we give a few examples of natural convex functions with huge $\frac{L}{\mu}$. One very simple example is

$$f(x) = \frac{1}{2}x_1^2 + \frac{1}{2}\sum_{k=1}^{n-1}(x_k - x_{k+1})^2 + \frac{1}{2}x_n^2$$

This problem appears everywhere from physical systems, graph problems, signal denoising to statistics. For this problem, we have $L = \Theta(1)$ and $\mu = \Theta(\frac{1}{n^2})$. This example shows that you can have large $\frac{L}{\mu}$ even with small description coefficients in the problem. Here is another example:

$$f(x) = \|Ax - b\|_2^2$$

where $A$ is a random $n \times n$ matrix with independent Gaussian entries. For this problem, we have $L = \Theta(n)$ and $\mu = \Theta(\frac{1}{n})$ with high probability. For an extreme example, the function $|x|$ has both $L = +\infty$ and $\mu = 0$.

### Accuracy of analysis

For gradient descent and accelerated gradient descent, the convergence rate is asymptotically accurate if we take $L \sim \lambda_{\max}(\nabla^2 f(x^*))$ and $\mu \sim \lambda_{\min}(\nabla^2 f(x^*))$ and $f \in \mathcal{C}^2$. This is because any $\mathcal{C}^2$ function locally looks like a quadratic around its minimum. We take gradient descent as an example. Note that gradient descent is invariant under any rotation and translation transformation. Therefore, we can assume $x^* = 0$ and that $\nabla^2 f(x^*)$ is a diagonal matrix with diagonals $\lambda_1, \lambda_2, \cdots, \lambda_n$. When $x$ is close to the optimal $x^* = 0$, gradient descent $x^{(\text{new})} = x - h\nabla f(x)$ is similar to a coordinate-wise iteration $x_i^{(\text{new})} = (1 - h\lambda_i)x_i$. This gives a precise description of the convergence rate.

For more complicated algorithms, the theory of convergence rate and cost per iteration sometimes gives a bad estimate of the reality. For example, the interior point method often takes $O(\log n)$ iterations to converge despite its $\sqrt{n}$ bound on convergence rate; simplex method is often the best of choice for solving linear programs despite its exponential complexity bound.

The cost per iterations sometimes can be misleading also. Each iteration of interior point methods takes $\text{nnz}(A) + \text{rank}(A)^2$ in theory while it takes nearly linear time in practice for sparse matrices due to the effectiveness of Cholesky decomposition. The $k^{th}$ iteration of cutting plane methods often takes $nk + k^{O(1)}$ if implemented correctly despite its super-linear bound.

Finally, the error of floating point computation gradually starts to have an effect when the target accuracy $\epsilon$ is small.

## ■ 3.3 Relations Between Different First-order Methods

In the table 3.1, we omitted many cases. For example, what is the runtime if we have a stochastic function $\min_x \mathbf{E}_\omega f_\omega(x)$ such that $\mathbf{E}_\omega f_\omega(x)$ is strongly convex? What is the runtime if we have a finite sum problem $\frac{1}{n}\sum_i f_i(x)$

which is not strongly convex? In this section, we show that one can reduce runtime of one case to the another case. To highlight the idea, we will be informal and ignore polylogarithmic terms. We note that some of the reduction ideas are not just useful for proving theorem, but are practical tricks to speed up algorithm and to avoid the knowledge of parameters. See [37, 22, 50, 2] for more reduction tricks for first-order methods.

### From Strongly Convex to Convex

Suppose we have an algorithm $\mathcal{A}(f, x^{(0)})$ that inputs a $\mu$-strongly convex function $f$ such that $\|\nabla f(x)\|_2 \leq G$ for the domain of $f$ (such as a ball of radius $R$ around $x^{(0)}$)[1] and outputs a point $x$ such that $f(x) \leq \min_x f(x) + \epsilon$ in $\frac{G^2}{\mu\epsilon}$ calls to $\nabla f$. Equivalently, $f(x) \leq \min_x f(x) + \frac{G^2}{\mu k}$ in $k$ calls to $\nabla f$. Now, we show how to recover the result $(\frac{GR}{\epsilon})^2$ by the following algorithm:

- Output $\mathcal{A}(\widetilde{f}, x^{(0)})$ where $\widetilde{f}(x) = f(x) + \frac{G}{R\sqrt{2k}}\|x\|_2^2$.

To understand the result, we let $f_\sigma(x) = f(x) + \frac{\sigma}{2}\|x\|_2^2$ and let $x$ be the output of $\mathcal{A}(f_\sigma)$. Note that $f_\sigma$ is $\sigma$ strongly convex. Hence, in $k$ iterations, we have that

$$f_\sigma(x) \leq \min_x f_\sigma(x) + \frac{G^2}{\sigma k} \leq f_\sigma(x^*) + \frac{G^2}{\sigma k} = f(x^*) + \frac{\sigma}{2}R^2 + \frac{G^2}{\sigma k}$$

where $x^*$ is the minimum of $f(x)$. To get the best runtime for the fix $k$, we choose $\sigma$ such that

$$\frac{\sigma}{2}R^2 = \frac{G^2}{\sigma k}$$

which leads to $\sigma = \sqrt{\frac{2}{k}}\frac{G}{R}$ and hence the total error is $\sqrt{2}\frac{GR}{\sqrt{k}}$ which is exactly what we expect to get (up to constant).

### From Convex to Strongly Convex

Suppose we have an algorithm $\mathcal{A}(f, x^{(0)})$ that inputs a Lipschitz function $f$ such that $\|\nabla f(x)\|_2 \leq G$ and that $\|x^{(0)} - x^*\|_2 \leq R$ and outputs a point $x$ such that $f(x) \leq \min_x f(x) + \epsilon$ in $(\frac{GR}{\epsilon})^2$ calls to $\nabla f$. Equivalently, $f(x) \leq \min_x f(x) + \frac{GR}{\sqrt{k}}$ in $k$ calls to $\nabla f$. Now, we show how to recover the result $\widetilde{O}(\frac{G^2}{\mu\epsilon})$ by the following algorithm:

- For $i = 1, 2, \cdots, \widetilde{O}(1)$
  - $x^{(i)} \leftarrow \mathcal{A}(f, x^{(i-1)})$ for $\frac{G^2}{\mu\epsilon}$ iterations

By the guarantee of $\mathcal{A}$, we have that

$$f(x^{(i)}) \leq \min_x f(x) + \frac{G\|x^{(i-1)} - x^*\|_2}{\sqrt{k}} = \min_x f(x) + \sqrt{\mu\epsilon}\|x^{(i-1)} - x^*\|_2. \tag{3.3.1}$$

On the other hand, the strong convexity of $f$ shows that $f(x^{(i)}) - \min_x f(x) \geq \frac{\mu}{2} \cdot \|x^{(i)} - x^*\|_2^2$. Therefore, we have that

$$\|x^{(i)} - x^*\|_2^2 \leq 2\sqrt{\frac{\epsilon}{\mu}}\|x^{(i-1)} - x^*\|_2.$$

Therefore, if $\|x^{(i-1)} - x^*\|_2 \geq 8\sqrt{\frac{\epsilon}{\mu}}$, then the distance to OPT decreases by at least a factor of two. After $\widetilde{O}(1)$ iterations, we have that $\|x^{(i-1)} - x^*\|_2 = O\left(\sqrt{\frac{\epsilon}{\mu}}\right)$ and hence (3.3.1) shows that $f(x^{(i)}) \leq \min_x f(x) + \sqrt{\mu\epsilon} \cdot O\left(\sqrt{\frac{\epsilon}{\mu}}\right) = \min_x f(x) + O(\epsilon)$.

**Exercise 3.3.1.** Given an algorithm that can minimize convex $f$ with $\epsilon$ additive error in $\sqrt{\frac{LR^2}{\epsilon}}$ iterations, show how to use this algorithm to minimize $f$ again with $\epsilon$ additive error in $\widetilde{O}\left(\sqrt{\frac{L}{\mu}}\log\left(\frac{LR^2}{\epsilon}\right)\right)$ iterations.

[1]Any strongly convex function on $\mathbb{R}^n$ cannot be uniformly Lipschitz on $\mathbb{R}^n$. Therefore, we must assume the domain is bounded in some way.

## ■ 3.4 Equivalences between Oracles

### Oracles for Convex Sets

Grötschel, Lovasz and Schrijver [23] introduced five different ways to access convex sets, showed they are equivalent and used them to get polynomial-time algorithms for a variety of combinatorial problems (including the first ones in many cases). Here are four basic oracles for a convex set $K \subseteq \mathbb{R}^n$. The original definitions are more general, allowing for error parameters in each oracle, i.e., approximate versions of all the oracles below. Here we will focus on exact oracles for simplicity. We also omit the fifth oracle VIOL, which checks whether the convex set satisfies a given inequality or gives a violating point in the convex set, since this is equivalent to the OPT oracle below up to a logarithmic factor.

**Definition 3.4.1** (Membership Oracle (MEM))**.** Queried with a vector $y \in \mathbb{R}^n$, the oracle either

- asserts that $y \in K$, or
- asserts that $y \notin K$.

**Definition 3.4.2** (Separation Oracle (SEP))**.** Queried with a vector $y \in \mathbb{R}^n$, the oracle either

- asserts that $y \in K$, or
- finds a unit vector $c \in \mathbb{R}^n$ such that $c^T x \leq c^T y$ for all $x \in K$.

**Definition 3.4.3** (Validity Oracle (VAL))**.** Queried with a unit vector $c \in \mathbb{R}^n$, the oracle either[2]

- outputs $\max_{x \in K} c^\top x$, or
- asserts that $K$ is empty.

**Definition 3.4.4** (Optimization Oracle (OPT))**.** Queried with a unit vector $c \in \mathbb{R}^n$, the oracle either

- finds a vector $y \in \mathbb{R}^n$ such that $y \in K$ and $c^T x \leq c^T y$ for all $x \in K$, or
- asserts that $K$ is empty.

According to the definition, the separation oracle gives strictly more information than the membership oracle and the optimization oracle gives more information than the validity oracle. Depending on the problem, usually one of the oracles will be the preferred way to access. For example, for the polytope given by $\{Ax \geq b\}$, the separation oracle is the preferred way because the membership oracle takes as much time as the separation oracle, and both validity and optimization involve solving a linear program. On the contrary, the preferred oracle for the convex set $\mathrm{conv}(\{a_i\})$ is the optimization oracle because $\max_{x \in \mathrm{conv}(\{a_i\})} \theta^\top x$ can be solved by checking $x = a_i$ for each $i$. In combinatorial optimization, many polytopes have exponentially many vertices and constraints but one can use combinatorial structure to solve the optimization problem.

**Example 3.4.5.** The spanning tree polytope of an undirected graph $G = (V, E)$ is given by

$$P = \{x \in \mathbb{R}^E_{\geq 0} | \sum_{e \in E} x_e = |V| - 1, \quad \sum_{(u,v) \in E \cap (S \times S)} x_{(u,v)} \leq |S| - 1 \quad \forall S \subseteq V\}.$$

[2] We use a slightly different definition than [23] for clarity.

$$\begin{tikzcd}
OPT(K) = \partial\delta_{K^*} \arrow[r, dashed, shift right] \arrow[d, shift right] & SEP(K) = \partial\delta_K \arrow[l, dashed, shift right] \arrow[d, shift right] \\
VAL(K) = \delta_{K^*} \arrow[u, dashed, shift right] & MEM(K) = \delta_K \arrow[u, dashed, shift right]
\end{tikzcd}$$

$\longrightarrow$ is $\tilde{O}(1)$

$\dashrightarrow$ is $\tilde{O}(n)$

**Figure 3.4.1:** The relationships among the four oracles for convex sets.

## Oracles for Convex Functions

Now, we generalize the oracles for convex sets to convex functions. The membership oracle and separation oracle can generalized as follows

**Definition 3.4.6** (Evaluation Oracle (EVAL))**.** Queried with a vector $y$, the oracle outputs $f(y)$.

**Definition 3.4.7** (Subgradient Oracle (GRAD))**.** Queried with a vector $y$, the oracle outputs $f(y)$ and a vector $g \in \mathbb{R}^n$ such that

$$f(x) \geq f(y) + g^\top (x - y) \text{ for all } x \in \mathbb{R}^n \tag{3.4.1}$$

To generalize the validity oracle, we define the convex (Fenchel) conjugate of $f$:

**Definition 3.4.8** (Convex Conjugate)**.** For any function $f$, we define the convex conjugate

$$f^*(\theta) \stackrel{\text{def}}{=} \sup_{x \in \mathbb{R}^n} \theta^\top x - f(x).$$

Note that $f^*$ is convex because it is the maximum of linear functions. Also, we have $f^*(0) = \inf_x f(x)$. Note that[3] $\delta_K^*(c) = \sup_{x\in K} c^\top x$. Therefore, the validity oracle is simply the evaluation oracle for $\delta_K^*$. The following lemma shows that the optimization oracle is simply the gradient oracle for $\delta_K^*$.

**Lemma 3.4.9.** *For any continuously differentiable function $f$, we have that $\nabla f^*(\theta) = \arg\max_x \theta^\top x - f(x)$.*

*Proof.* Let $x_\theta = \arg\sup_x \theta^\top x - f(x)$. By definition, we have that $f^*(\theta) = \theta^\top x_\theta - f(x_\theta)$ and that $f^*(\eta) \geq \eta^\top x_\theta - f(x_\theta)$ for all $\eta$. Therefore,

$$f^*(\eta) \geq f^*(\theta) + x_\theta^\top (\eta - \theta) \text{ for all } \eta.$$

Therefore, $\nabla f^*(\theta) = x_\theta$. □

Note that this lemma shows that $\nabla \delta_K^*(\theta) = \arg\max_{x\in K} \theta^\top x$. So, the gradient oracle for $\delta_K^*$ is exactly the optimization oracle for $K$.

Figure 3.4.2 shows the current best reduction between these four function oracles. Note that according to our definition of the GRAD oracle, it also outputs the function value (EVAL). It is not hard to show that you can use $n+1$ calls to evaluation oracle to compute one gradient (using finite difference) and that you can use $\tilde{O}(n)$ calls to gradient oracle of $f$ to compute one gradient of $f^*$ (using cutting plane method). In the next section, we will show the rest of the reduction. However, it is still open whether all of these reductions are tight:

**Open Problem.** Prove that it takes $\Omega(n^2)$ calls to the evaluation oracle of $f$ to compute the gradient of $f^*$.

## Convex Conjugate and Equivalences

Here are some examples of conjugates.

[3]Recall that $\delta_C(x) = 0$ if $x \in C$ and $+\infty$ otherwise.

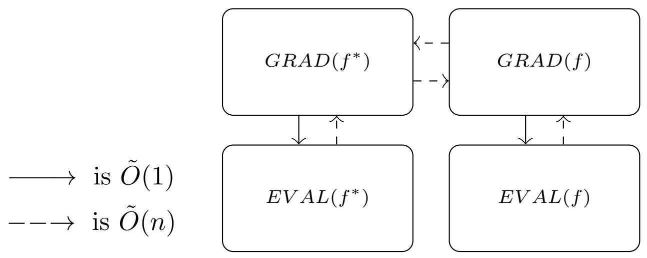

$GRAD(f^*)$
$GRAD(f)$
$EVAL(f^*)$
$EVAL(f)$
$\longrightarrow$ is $\tilde{O}(1)$
$--\rightarrow$ is $\tilde{O}(n)$

**Figure 3.4.2:** This illustrates the relationships of oracles for a convex function $f$ and its convex conjugate $f^*$.

**Exercise 3.4.10.** The conjugate of $f(x) = \frac{1}{p}\|x\|_p^p$ is $f^*(\theta) = \frac{1}{q}\|\theta\|_q^q$ where $\frac{1}{p} + \frac{1}{q} = 1$. The conjugate of $f(x) = \langle a, x\rangle - b$ is $f^*(\theta) = \begin{cases} b, & \theta = a \\ +\infty & \text{otherwise} \end{cases}$. The conjugate of $f(x) = \sum_i e^{x_i}$ is $f^*(\theta) = \begin{cases} \sum_i \theta_i \log \theta_i - \theta_i & \text{if } \theta_i \geq 0 \text{ for all } i \\ +\infty & \text{otherwise} \end{cases}$.

**Exercise 3.4.11.** For any $x, \theta$, we have $\theta^\top x \leq f(x) + f^*(\theta)$.

**Exercise 3.4.12.** Prove that the gradient of $f$ is $L$-Lipschitz if and only if $f^*$ is $\frac{1}{L}$ strongly convex.

The following lemma shows that indeed one can recover $f$ from $f^*$.

**Lemma 3.4.13** (Involution property)**.** *For any convex function $f$ with a closed epigraph, we have that $f^{**} = f$.*

*Proof.* Since epi$f$ is closed and convex, Corollary 1.2.2 shows that it is an intersection of halfspaces, i.e.,

$$f(x) = \sup_{\{\theta, b\} \in \mathcal{H}} \theta^\top x - b$$

where $\mathcal{H}$ is the set of supporting planes of epi$f$ and contains all affine lower bounds on $f$, namely, $f(x) \geq \theta^\top x - b$ for all $x$. For a fixed $\theta$, any feasible $b$ satisfies $b \geq \theta^\top x - f(x)$ for all $x$. So, the smallest feasible value satisfies

$$b = \sup_x \theta^\top x - f(x) = f^*(\theta).$$

Hence,

$$f(x) = \sup_{\{\theta, b\} \in \mathcal{H}} \theta^\top x - b = \sup_\theta \theta^\top x - f^*(\theta) = f^{**}(x).$$

□

Since we can use $\nabla f$ to compute $\nabla f^*$ (via cutting plane method), the involution property shows that we can do the reverse – use $\nabla f^*$ to compute $\nabla f$. Going back to the example about conv($\{a_i\}$), since we know how to compute $\max_{x \in \text{conv}(\{a_i\})} \theta^\top x = \delta^*_{\text{conv}(\{a_i\})}(\theta)$, this reduction gives us a way to separate conv($\{a_i\}$), or equivalently, to compute the (sub)gradient of $\delta^*_{\text{conv}(\{a_i\})}$. In combinatorial optimization, many convex sets are given by the convex hull for some discrete objects. In many cases, the only known way to do the separation is via such reductions.

Recall that for any linear space $X$, $X^*$ denotes the dual space, i.e., the set of all linear functions on $X$ and that under mild assumptions, we have $X^{**} = X$. Therefore, there are two natural "coordinate systems" to record a convex function, the primal space $X$ and the dual space $X^*$. Under these "coordinate systems", we have the dual functions $f$ and $f^*$.

**Exercise 3.4.14** (Order reversal)**.** $f \geq g \iff f^* \leq g^*$.

Interestingly, convex conjugate is the unique transformation on convex functions that satisfies involution and order reversal.

**Theorem 3.4.15** ([5])**.** *Given a transformation $\mathcal{T}$ that maps the set of lower-semi-continuous convex functions onto itself such that $\mathcal{T}\mathcal{T}\phi = \phi$ and $\phi \leq \psi \implies \mathcal{T}\phi \geq \mathcal{T}\psi$ for all lower-semi-continuous convex functions $\phi$ and $\psi$. Then, $\mathcal{T}$ is essentially the convex conjugate, namely, there is an invertible symmetric linear transformation $B$, a vector $v_0$ and a constant $C_0$ such that*

$$(\mathcal{T}\phi)(x) = \phi^*(Bx + v_0) + v_0^\top x + C_0.$$

When the function $f$ is in $\mathcal{C}^1$, we can approximate the gradient of a function by calculating a finite difference

$$\frac{\partial f}{\partial x_i} = \frac{f(x+he_i) - f(x)}{h} + o(1)$$

which only takes $n+1$ calls to the evaluation oracle (for computing $f(x), f(x+he_1), \cdots, f(x+he_n)$). The only issue is that the convex function may not be differentiable. However, any convex Lipschitz function is twice differentiable almost everywhere (see the proof below). Therefore, we can simply perturb $x$ with random noise, then apply a finite difference. To see the idea more precisely, we first observe that the norm of the Hessian can be bounded in expectation for a Lipschitz function.

**Lemma 3.4.16.** *For any $L$-Lipschitz convex function $f$ defined in a unit ball, we have $\mathbf{E}_{x\in B(0,1)}\|\nabla^2 f(x)\|_F \leq nL$.*

*Proof.* Since $\nabla^2 f \succeq 0$, we have $\|\nabla^2 f(x)\|_F \leq \mathrm{tr}\nabla^2 f(x)$. Therefore, that

$$\int_{B(0,1)} \|\nabla^2 f(x)\|_F dx \leq \int_{B(0,1)} \mathrm{tr}\nabla^2 f(x) dx = \int_{B(0,1)} \Delta f(x) dx.$$

Using Stokes' Theorem, we have

$$\int_{B(0,1)} \Delta f(x) dx = \int_{\partial B(0,1)} \langle \nabla f(x), n(y)\rangle \, dy \leq |\partial B(0,1)| \cdot L.$$

Hence, we have

$$\mathbf{E}_{x\in B(0,1)}\|\nabla^2 f(x)\|_F \leq \frac{|\partial B(0,1)|}{|B(0,1)|} L = nL.$$

□

To turn this into an algorithm, we need to develop this a bit further.

**Lemma 3.4.17** ([33])**.** *Let $B_\infty(x,r) = \{y : \|x-y\|_\infty \leq r\}$. For any $0 < r_2 \leq r_1$ and any convex function $f$ defined on $B_\infty(x, r_1 + r_2)$ with $\|\nabla f(z)\|_\infty \leq L$ for any $z \in B_\infty(x, r_1+r_2)$ we have*

$$\mathbf{E}_{y\in B_\infty(x,r_1)} \mathbf{E}_{z\in B_\infty(y,r_2)} \|\nabla f(z) - g(y)\|_1 \leq n^{3/2}\frac{r_2}{r_1} L$$

*where $g(y)$ is the average of $\nabla f$ over $B_\infty(y, r_2)$.*

*Proof.* Let $\omega_i(z) = \langle \nabla f(z) - g(y), e_i\rangle$ for all $i \in [n]$. Then, we have that

$$\int_{B_\infty(y,r_2)} \|\nabla f(z) - g(y)\|_1 \, dz \leq \sum_i \int_{B_\infty(y,r_2)} |\omega_i(z)| \, dz.$$

Since $\int_{B_\infty(y,r_2)} \omega_i(z) dz = 0$, the Poincare inequality for a box (Theorem 3.4.20 below) shows that

$$\begin{aligned}\int_{B_\infty(y,r_2)} |\omega_i(z)| \, dz &\leq r_2 \int_{B_\infty(y,r_2)} \|\nabla \omega_i(z)\|_2 \, dz \\ &= r_2 \int_{B_\infty(y,r_2)} \left\|\nabla^2 f(z) e_i\right\|_2 dz \\ \sum_i \int_{B_\infty(y,r_2)} |\omega_i(z)| \, dz &\leq \sqrt{n} r_2 \int_{B_\infty(y,r_2)} \left\|\nabla^2 f(z)\right\|_F dz\end{aligned}$$

Since $f$ is convex, we have that $\left\|\nabla^2 f(z)\right\|_F \leq \mathrm{tr}\nabla^2 f(z) = \Delta f(z)$. Therefore, we have

$$\begin{aligned}\mathbf{E}_{z\in B_\infty(y,r_2)} \|\nabla f(z) - g(y)\|_1 \, dz &\leq \sqrt{n} r_2 \mathbf{E}_{z\in B_\infty(y,r_2)} \Delta f(z) dz \\ &= \sqrt{n} r_2 \Delta h(y)\end{aligned}$$

where $h = \frac{1}{(2r_2)^n} f * \chi_{B_\infty(0,r_2)}$ where $\chi_{B_\infty(0,r_2)}$ is 1 on the set $B_\infty(0,r_2)$ and 0 on outside.

Integrating by parts, we have that

$$\int_{B_\infty(x,r_1)} \Delta h(y)dy = \int_{\partial B_\infty(x,r_1)} \langle \nabla h(y), n(y) \rangle \, dy$$

where $\Delta h(y) = \sum_i \frac{d^2 h}{dx_i^2}(y)$ and $n(y)$ is the normal vector on $\partial B_\infty(x,r_1)$ the boundary of the box $B_\infty(x,r_1)$, i.e. standard basis vectors. Since $f$ is $L$-Lipschitz with respect to $\|\cdot\|_\infty$ so is $h$, i.e. $\|\nabla h(z)\|_\infty \le L$. Hence, we have that

$$\mathbf{E}_{y\in B_\infty(x,r_1)}\Delta h(y) \le \frac{1}{(2r_1)^n}\int_{\partial B_\infty(x,r_1)} \|\nabla h(y)\|_\infty \|n(y)\|_1 \, dy \le \frac{1}{(2r_1)^n}\cdot 2n(2r_1)^{n-1}\cdot L = \frac{nL}{r_1}.$$

Therefore, we have that

$$\mathbf{E}_{y\in B_\infty(x,r_1)}\mathbf{E}_{z\in B_\infty(y,r_2)} \|\nabla f(z) - g(y)\|_1 \, dz \le n^{3/2}\frac{r_2}{r_1}L.$$

□

This lemma shows that we can implement an approximate gradient oracle (GRAD) using an evaluation oracle (EVAL) even for non-differentiable functions. By the involution property again, this completes all the reductions in Figure 3.4.2. This argument can also be used to construct an approximate separation oracle for a convex set from a membership oracle.

**Theorem 3.4.18.** *Let $K$ be a convex body s.t. $B(0,1) \subseteq K \subseteq B(0,R)$. Then, for any $\eta \in (0,0.5)$,we can compute an $\eta$-approximate separation oracle for $K$ using $O(n\log(nR/\eta))$ queries to a membership oracle for $K$.*

By an approximate separation oracle we mean that either the queried point $x$ lies within distance $\eta$ of $K$ or the oracle provides a halfspace $c^T y \le c^T x + \eta$ for all points $y \in K$ at distance at least $\eta$ from the boundary of $K$. We sketch the proof here in a sentence: to separate $x$ from the set $K$, we consider the convex function $h_x : K \to \mathbb{R}$ defined as follows:

$$h_x(y) = -\max\{\alpha : d + \alpha x \in K\}$$

and use Lemma 3.4.17 to compute an approximate gradient of $h$ at the origin. For more details, including dependence on approximation parameters, see [33].

**Exercise 3.4.19.** What is the best possible bound in Lemma 3.4.17?

In the proof of Lemma 3.4.17, we used the following fact applied to the case when $\Omega$ is a cube. In this case, the coefficient on the RHS is given by the Cheeger or KLS constant of the cube and is 1. This is an example of an isoperimetric inequality. Such inequalities will play an important role in this book.

**Theorem 3.4.20** ($L^1$Poincare inequality)**.** *Let $\Omega$ be connected, bounded and open. Then the following (best-possible) inequality holds for any smooth function $f : \Omega \to \mathbb{R}$:*

$$\left\| f - \frac{1}{|\Omega|}\int_\Omega f(x)\,dx \right\|_{L^1(\Omega)} \le \left( \sup_{S\subset\Omega} \frac{2|S||\Omega\setminus S|}{|\partial S||\Omega|} \right) \|\nabla f\|_{L^1(\Omega)}$$

*where the supremum is over all subsets $S$ s.t. $S$ and $\Omega \setminus S$ are both connected.*

**Exercise 3.4.21.** Prove the inequality in Theorem 3.4.20 using the classical coarea formula.

## ■ 3.5 Optimization from Membership via Sampling

In this chapter, we assumed the oracle can be computed exactly. However, it is still open to determine the best runtime for reductions between noisy oracles. In particular, the question about minimizing (approximately) convex functions under noisy oracles is an active research area and the gaps between the lower bound and upper bound for many problems are still quite large ([8, 19, 10] based on [24]). We will see these methods in detail later. For now, to put them in the context of oracles, we just discuss the following theorem.

**Theorem 3.5.1** (OPT via sampling)**.** *Let $F(x) = e^{-\alpha c^T x} 1_K(x)$ for some convex set in $\mathbb{R}^n$ and vector $c \in \mathbb{R}^n$. Suppose $\min_{x \in K} c^T x$ is bounded. Let $x$ be a random sample from the distribution with density proportional to $F(x)$. Then,*

$$\mathbf{E}\left(c^T x\right) \leq \min_K c^T x + \frac{n}{\alpha}.$$

*Proof.* We will show that the worst case is when the convex set $K$ is an infinite cone. WLOG assume that the minimum is at $x = 0$. Replace every cross-section of $K$ along $c$ with an $(n-1)$-dimensional ball of the same volume as the cross-section. This does not affect the expectation. Suppose the expectation is $\mathbf{E}c^T x = a$. Then replace the part of $K$ with $c^T x \leq a$ with a cone whose base is the cross-section at $a$ and apex is at zero. This only makes the expectation larger. Next, replace the all of $K$ to the right of $a$ with the infinite conical extension of the cone to the left of $a$. Again, this expectation can only be higher.

Now for this infinite cone, we compute, using $y = c^T x$,

$$\begin{aligned} \mathbf{E}\left(c^T x\right) &= \frac{\int_0^\infty y e^{-\alpha y} y^{n-1}\, dy}{\int_0^\infty e^{-\alpha y} y^{n-1}\, dy} \\ &= \frac{1}{\alpha} \frac{\int_0^\infty e^{-y} y^n\, dy}{\int_0^\infty e^{-y} y^{n-1}\, dy} \\ &= \frac{1}{\alpha} \frac{n!}{(n-1)!} = \frac{n}{\alpha}. \end{aligned}$$

□

The theorem says that if we sample according to $e^{-\alpha c^T x}$ for $\alpha = n/\varepsilon$, we will get an $\varepsilon$-approximation to the optimum. However, sampling from such a density is not trivial. Instead, we will have to go through a sequence of overlapping distributions, starting with one that is easy to sample and the last one being such a distribution that is focused close to the minimum. The complexity of sampling is polynomial in the dimension and in a suitable notion of probabilistic distance between the starting distribution and the target distribution. The sampling algorithm only uses a membership (EVAL) oracle.

**Exercise 3.5.2.** Extend Theorem 3.5.1 by replacing $c^T x$ with any convex function $f(x)$.

**Open Problem.** Given an approximately convex function $F$ on unit ball such that $\max_{\|x\|_2 \leq 1} |f(x) - F(x)| \leq \varepsilon/n$ for some convex function $f$, how efficiently can we find $x$ in the unit ball such that $F(x) \leq \min_{\|x\|_2 \leq 1} F(x) + O(\varepsilon)$? The current fastest algorithm takes $O(n^{4.5} \log^{O(1)}(n/\varepsilon))$ calls to the noisy EVAL oracle for $F$.

## ■ 3.6 Composite Problem via Duality

### Motivation

In this section, we will discuss a few algorithmic applications of duality. Suppose we have a "difficult" convex problem $\min_x f(x)$. Often, we can decompose the difficult problem as $f(x) = g(x) + h(Ax)$ such that $\nabla g^*(x)$ and $\nabla h^*(x)$ are easy to compute. We can compute its dual as follows:

$$\begin{aligned} \min_x g(x) + h(Ax) &= \min_x \max_\theta g(x) + \theta^\top Ax - h^*(\theta) \\ &= \max_\theta \min_x g(x) + (A^\top \theta)^\top x - h^*(\theta) \\ &= \max_\theta -g^*(-A^\top \theta) - h^*(\theta) \\ &= -\min_\theta g^*(-A^\top \theta) + h^*(\theta) \end{aligned}$$

where we used $h = h^{**}$ in the first line, the following minimax theorem on the second line, and the definition of $g^*$ on the third line.

**Theorem 3.6.1** (Sion's minimax theorem). *Let $X \subset \mathbb{R}^n$ be a compact convex set and $Y \subset \mathbb{R}^m$ be a convex set. If $f : X \times Y \to \mathbb{R} \cup \{+\infty\}$ such that $f(x, \cdot)$ is upper semi-continuous and quasi-concave on $Y$ for all $x \in X$ and $f(\cdot, y)$ is lower semi-continuous and quasi-convex on $X$ for all $y \in Y$. Then, we have*

$$\min_{x\in X}\sup_{y\in Y} f(x,y) = \sup_{y\in Y}\min_{x\in X} f(x,y).$$

*Remark.* Compactness is necessary. Consider $f(x, y) = x + y$.

We call "$g(x) + h(Ax)$" the primal problem and "$g^*(-A^\top\theta) + h^*(\theta)$" the dual problem. Often, the dual problem gives us some insight on the primal problem. However, we note that there are many ways to split a problem into two and hence many dual problems.

As an example, consider the unit capacity flow problem on a graph $G = (V, E)$:

$$\max_{Af=d,-1\le f\le 1} c^\top f$$

where $f \in \mathbb{R}^E$ is the flow vector, $A \in \mathbb{R}^{V\times E}$ encodes the flow conservation constraints, $d$ is the demand vector and $c$ is the cost vector. We can take the dual as follows:

$$\begin{aligned}\max_{Af=d,-1\le f\le 1} c^\top f &= \max_{-1\le f\le 1}\min_{\phi} c^\top f - \phi^\top(Af-d)\\ &= \min_{\phi}\max_{-1\le f\le 1} \phi^\top d + (c - A^\top\phi)^\top f\\ &= \min_{\phi} \phi^\top d + \sum_{e\in E} |c - A^\top\phi|_e.\end{aligned}$$

For the case $c = 0$ and $d = F \cdot 1_{st}$ (the maximum flow problem), the dual problem is the minimum $s-t$ cut problem with the cut given by $\{v \in V \text{ such that } \phi(v) \ge t\}$. Note that there are $|E|$ variables in primal and $|V|$ variables in dual. So, in some sense, the dual problem is easier for dense graphs.

Although we do not have a way to turn a minimum $s-t$ cut to a maximum $s-t$ flow in general, we will see various tricks to reconstruct the primal solution from the dual solution by modifying the problem.

## Example: Semidefinite Programming

When you can solve the dual problem, the proof of the optimality of the dual problem often directly gives you the solution of the primal problem, or gives you some way to solve the primal problem efficiently. Here, we use the semidefinite programming as an concrete example. Consider the semidefinite programming (SDP) problem:

$$\max_{X\succeq 0} C \bullet X \text{ s.t. } A_i \bullet X = b_i \text{ for } i = 1, 2, \cdots, m \tag{3.6.1}$$

and its dual

$$\min_{y} b^\top y \text{ s.t. } \sum_{i=1}^m y_i A_i \succeq C \tag{3.6.2}$$

where $X$, $C$, $A_i$ are $n \times n$ symmetric matrices and $b, y \in \mathbb{R}^m$. SDP is a generalization of linear programming and is useful for various problems involving matrices. If we apply the current-best cutting plane method naively on the primal problem, we would get $\tilde{O}(n^2(Z + n^4))$ time algorithm for the primal (because there are $n^2$ variables) and $\tilde{O}(m(Z + n^\omega + m^2))$ for the dual where $Z$ is the total number of non-zeros in $A_i$. (The term $Z + n^\omega$ is the cost of computing $\sum y_i A_i$ and finding its minimum eigenvalue.) Generally, $n^2 \gg m$ and hence it takes much less time to solve the dual.

We note that

$$\min_{\sum_{i=1}^m y_iA_i\succeq C} b^\top y = \min_{v^\top(\sum_{i=1}^m y_iA_i - C)v\ge 0\ \forall\|v\|_2=1} b^\top y.$$

In each step of the cutting plane method, the (sub)gradient oracle either outputs $b$ or outputs one of the cutting planes

$$v^\top(\sum_{i=1}^m y_iA_i - C)v \ge 0.$$

Let $S$ be the set of all cutting planes used in the algorithm. Then, the proof of the cutting plane method shows that

$$\min_{\sum_{i=1}^m y_i A_i \succeq C} b^\top y = \min_{v^\top(\sum_{i=1}^m y_i A_i - C)v \geq 0 \ \forall v \in \mathcal{S}} b^\top y \pm \varepsilon. \tag{3.6.3}$$

The key idea to obtaining the primal solution is to take the dual of the right hand side (which is an approximate dual of the original problem). Now, we have

$$\begin{aligned}\min_{v^\top(\sum_{i=1}^m y_i A_i - C)v \geq 0 \ \forall v \in \mathcal{S}} b^\top y &= \min_y \max_{\lambda_v \geq 0} b^\top y - \sum_{v \in \mathcal{S}} \lambda_v v^\top (\sum_{i=1}^m y_i A_i - C) v \\ &= \max_{\lambda_v \geq 0} \min_y C \bullet \sum_{v \in \mathcal{S}} \lambda_v v v^\top + b^\top y - \sum_{i=1}^m y_i \sum_{v \in \mathcal{S}} \lambda_v v v^\top \bullet A_i \\ &= \max_{X = \sum_{v \in \mathcal{S}} \lambda_v v v^\top, \lambda_v \geq 0} \min_y C \bullet X + \sum_{i=1}^m y_i (b_i - X \bullet A_i) \\ &= \max_{X = \sum_{v \in \mathcal{S}} \lambda_v v v^\top, \lambda_v \geq 0, X \bullet A_i = b_i} C \bullet X.\end{aligned}$$

Note that this is exactly the primal SDP problem, except that we restrict the set of solutions to the form $\sum_{v \in \mathcal{S}} \lambda_v v v^\top$ with $\lambda_v \geq 0$. Also, we can write this problem as a linear program:

$$\max_{\sum_v \lambda_v v^\top A_i v = b_i \text{ for all } i, \lambda_v \geq 0} \sum_v \lambda_v v^\top C v. \tag{3.6.4}$$

Therefore, we can simply solve this linear program and recover an approximate solution for SDP $\sum_{v \in \mathcal{S}} \lambda_v v v^\top$. By (3.6.3), we know that this is an approximate solution with the same guarantee as the dual SDP.

Now, we analyze the runtime of this algorithm. This algorithm contains two phases: solve the dual SDP via cutting plane method, and solve the primal linear program. Note that each step of the cutting plane method involves finding a separation hyperplane of $\sum_{i=1}^m y_i A_i \succeq C$.

**Exercise 3.6.2.** Let $\Omega = \{y \in \mathbb{R}^m : \sum_{i=1}^m y_i A_i \succeq C\}$. Show that one can implement the separation oracle in time $O^*(Z + n^\omega)$ via eigenvalue computation.

Therefore, the first phase takes $O^*(m(Z + n^\omega + m^2))$ time in total. Since the cutting plane method takes $O^*(m)$ steps, we have $|\mathcal{S}| = O^*(m)$. In the second phase, we need to solve a linear program (3.6.4) with $O^*(m)$ variables with $O(m)$ constraints. It is known how to solve such linear programs in time $O^*(m^{2.38})$ [14]. Hence, the total cost is dominated by the first phase

$$O^*\left(mZ + mn^\omega + m^3\right).$$

**Open Problem 3.6.3.** In the first phase, each step involves computing an eigenvector of similar matrices. So, can we use matrix update formulas to decrease the cost per step in the cutting plane to $O^*(Z + n^2)$? Namely, can we solve SDP in time $O^*\left(mZ + m^3\right)$?

## Duality and Convex Hull

The $\min_x g(x) + h(Ax)$ in general can be solved by a similar trick. To make the geometric picture clear, we consider its linear optimization version: $\min_{(x,t_1) \in \text{epi}\, g, (Ax, t_2) \in \text{epi}\, h} t_1 + t_2$. To simplify the notation, we consider the problem

$$\min_{x \in K_1, Mx \in K_2} c^\top x$$

where $M \in \mathbb{R}^{m \times n}$, and we have convex sets $K_1 \subset \mathbb{R}^n$ and $K_2 \subset \mathbb{R}^m$.

To be concrete, let us consider the following example. Let $V_1$ be a set of students, $V_2$ be a set of schools. Each edge $e \in E$ represents a possible choice of a student. Let $w_e$ be the happiness of school/student if the student is assigned to that school. Suppose that every student can only be assigned to one school and school $b$ can accept $c_b$ students. Then, the problem can be formulated as

$$\min_{x_e \geq 0} \sum_{e \in E} w_e x_e \text{ subject to } \sum_{(a,b) \in E} x_{(a,b)} \leq 1 \ \forall a \in V_1, \ \sum_{(a,b) \in E} x_{(a,b)} \leq c_b \ \forall b \in V_2.$$

This is the weighted b-matching problem. Typically, the number of students is much more than the number of schools. Therefore, an algorithm with running time linear in the number of students is preferable. To apply our framework, we let

$$K_1 = \{x \in \mathbb{R}^E, x_e \geq 0, \sum_{(a,b)\in E} x_{(a,b)} \leq 1 \ \forall a \in V_1\},$$
$$K_2 = \{y \in \mathbb{R}^{V_2}, y_b \leq c_b\},$$

and $M \in \mathbb{R}^E \to \mathbb{R}^{V_2}$ is the map $(Tx)_b = \sum_{a:(a,b)\in E} x_{(a,b)}$.

To further emphasize its importance, consider some general examples here:

- Linear programming: $\min_{Ax=b,x\geq 0} c^\top x$: $K_1 = \{x \geq 0\}$, $K_2 = \{b\}$ and $M = A$.
- Semidefinite programming: $\min_{A_i \bullet X = b_i, X\succeq 0} C \bullet X$: $K_1 = \{X \succeq 0\}$, $K_2 = \{b\}$ and $M : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ defined by $(MX)_i = A_i \bullet X$.
- Matroid intersection: $\min_{x\in M_1\cap M_2} 1^\top x$: $K_1 = M_1$ and $K_2 = M_2$, $M = I$.
- Submodular minimization: $K_1 = \{y \in \mathbb{R}^n : \sum_{i\in S} y_i \leq f(S) \text{ for all } S \subset [n]\}$, $K_2 = \{y \leq 0\}$, $M = I$.
- Submodular flow: $K_1 = \{\varphi \in \mathbb{R}^E, \ell_e \leq \varphi_e \leq u_e \text{ for all } e \in E\}$, $K_2 = \{y \in \mathbb{R}^V : \sum_{i\in S} y_i \leq f(S) \text{ for all } S \subset [n]\}$, $M$ is the incidence matrix.

In all of these examples, it is easy to compute the gradient of $\ell^*_{K_1}$ and $\ell^*_{K_2}$. For the last three examples, it is not clear how to compute gradient of $\ell_{K_1}$ and/or $\ell_{K_2}$ directly. Furthermore, in all examples, $M$ maps from a larger space to the same or smaller space. Therefore, it is good to take advantage of the smaller space.

Before [34], the standard way is to use the equivalence of $\nabla\ell^*_{K_1}$ and $\nabla\ell_{K_1}$ , and apply cutting plane methods. With the running time of cutting plane methods, such an algorithm usually had theoretical running time at least $n^5$ and was of little practical value.

Now we rewrite the problem as we did in the beginning:

$$\begin{aligned}\min_{x\in K_1, Mx\in K_2} c^\top x &= \min_{x\in\mathbb{R}^n} c^\top x + \ell_{K_1}(x) + \ell_{K_2}(Mx)\\ &= \min_{x\in\mathbb{R}^n}\max_{\theta\in\mathbb{R}^m} c^\top x + \ell_{K_1}(x) + \theta^\top Mx - \ell^*_{K_2}(\theta)\\ &= \max_{\theta\in\mathbb{R}^m}\min_{x\in\mathbb{R}^n} c^\top x + \ell_{K_1}(x) + \theta^\top Mx - \ell^*_{K_2}(\theta)\\ &= \max_{\theta\in\mathbb{R}^m} -\ell^*_{K_1}(-c - M^\top\theta) - \ell^*_{K_2}(\theta).\end{aligned}$$

Taking the dual has two benefits. First, the number of variables is smaller. Second, the gradient oracle is something we can compute efficiently. Hence, cutting plane methods can be used to solve it in $O^*(m\mathcal{T} + m^3)$ where $\mathcal{T}$ is the time to evaluate $\nabla\ell^*_{K_1}$ and $\nabla\ell^*_{K_2}$. The only problem left is to recover the primal $x$.

The key observation is the following lemma:

**Lemma 3.6.4.** *Let $x_i \in K_1$ be the set of points output by the oracle $\nabla\ell^*_{K_1}$ during the cutting plane method. Define $y_i \in K_2$ similarly. Suppose that the cutting plane method ends with the guarantee that the additive error is less than $\varepsilon$. Then, we have that*

$$\min_{x\in K_1, Tx\in K_2} c^\top x \leq \min_{x\in\widetilde{K_1}, Tx\in\widetilde{K_2}} c^\top x \leq \min_{x\in K_1, Tx\in K_2} c^\top x + \varepsilon$$

*where $\widetilde{K_1} = \operatorname{conv}(x_i)$ and $\widetilde{K_2} = \operatorname{conv}(y_i)$.*

*Proof.* Let $\theta_i$ be the set of directions queried by the oracle for $\nabla\ell^*_{K_1}$ and $\varphi_i$ be the directions queried by the oracle for $\nabla\ell^*_{K_2}$. We claim that $x_i \in \nabla\ell^*_{\widetilde{K_1}}(\theta_i)$ and $y_i \in \nabla\ell^*_{\widetilde{K_2}}(\varphi_i)$. Having this, the algorithm cannot distinguish between $\widetilde{K_1}$ and $K_1$, and between $\widetilde{K_2}$ and $K_2$. Hence, the algorithm runs exactly the same, i.e., uses the same sequence of points. Therefore, we get the same value $c^\top x$. However, by the guarantee of cutting plane method, we have that

$$\min_{x\in\widetilde{K_2}, Tx\in\widetilde{K_2}} c^\top x \leq c^\top x \leq \min_{x\in K_1, Tx\in K_2} c^\top x + \varepsilon.$$

To prove the claim, we note that $x_i \in \nabla \ell^*_{\widetilde{K_1}}(\theta_i)$. Note that $\widetilde{K_1} \subset K_1$ and hence $\min_{x\in\widetilde{K_1}} \theta_i^\top x \geq \min_{x\in K_1} \theta_i^\top x$. Also, note that

$$x_i = \arg\min_{x\in K_1} \theta_i^\top x \in \widetilde{K_1}$$

and hence $\min_{x\in\widetilde{K_1}} \theta_i^\top x \leq \min_{x\in K_1} \theta_i^\top x$. Therefore, we have that $\min_{x\in\widetilde{K_1}} \theta_i^\top x = \min_{x\in K_1} \theta_i^\top x$. Therefore, $x_i \in \arg\min_{x\in K_1} \theta_i^\top x$. This proves the claim for $\widetilde{K_1}$. The proof for $\widetilde{K_2}$ is the same. □

This reduces the problem into the form $\min_{x\in\widetilde{K_1}, Tx\in\widetilde{K_2}} c^\top x$. For the second phase, we let $z_i = Mx_i \in \mathbb{R}^m$. Then, we have

$$\begin{aligned}\min_{x\in\widetilde{K_1}, Mx\in\widetilde{K_2}} c^\top x &= \min_{t_i\geq 0, s_i\geq 0, M\sum_i t_i x_i = \sum_i s_i y_i} c^\top(\sum_i t_i x_i) \\ &= \min_{t_i\geq 0, s_i\geq 0, \sum_i t_i z_i = \sum_i s_i y_i} \sum t_i \cdot c^\top x_i.\end{aligned}$$

Note that it takes $O^*(mZ)$ time to write down this linear program where $Z$ is the number of non-zeros in $M$. Next, we note that this linear program has $O^*(m)$ variables and $m$ constraints. Therefore, we can solve it in $O^*(m^{2.38})$ time.

Therefore, the total running time is

$$O^*(m(\mathcal{T} + m^2) + (mZ + m^{2.38})).$$

To conclude, we have the following theorem.

**Theorem 3.6.5.** *Given convex sets $K_1 \subset \mathbb{R}^n$ and $K_2 \subset \mathbb{R}^m$ with $m \leq n$ and a matrix $M : \mathbb{R}^n \to \mathbb{R}^m$ with $Z$ non-zeros. Let $\mathcal{T}$ be the cost to compute $\nabla \ell^*_{K_1}$ and $\nabla \ell^*_{K_2}$. Then, we can solve the problem*

$$\min_{x\in K_1, Mx\in K_2} c^\top x$$

*in time $O^*(m\mathcal{T} + mZ + m^3)$.*

*Remark.* We hid all sorts of terms in the log term hidden in $O^*$ such as the diameter of the set. Also this is the number of arithmetic operations, not the bit complexity.

Going back to the school/student problem, this algorithm gives a running time of

$$O^*(|V_2||E| + |V_2|^3)$$

which is linear in the number of students!

In general, this statement says that if we can split a convex problem into two parts, both easy to solve and one part has less variables, then we can solve it in time proportional to the smaller dimension.

**Exercise 3.6.6.** How fast we can solve the problem $\min_{x\in\cap_{i=1}^k K_i} c^\top x$ given the oracles $\nabla \ell^*_{K_i}$?

| Convex Problem | Assumption | Algorithm | Iter | Cost/Iter |
|---|---|---|---|---|
| $\min_x f(x)$ | - | Hybrid Center Cutting Plane | $n\log(\frac{n}{\epsilon})$ | $\mathcal{T}_f + n^2\log^{O(1)}(\frac{n}{\epsilon})$ |
| $\min_{x\in\Omega_1\cap\Omega_2} c^\top x$ | - | Hybrid Center Cutting Plane | $n\log(\frac{n}{\epsilon})$ | $\mathcal{T}_{\Omega_1}+\mathcal{T}_{\Omega_2}+n^2\log^{O(1)}(\frac{n}{\epsilon})$ |
| $\min_x f(x)+g(x)$ | $\\|\nabla f\\|_2\leq G,\ \\|x^*\\|_2\leq R$ | Gradient Descent | $(\frac{GR}{\epsilon})^2$ | $\mathcal{T}_{\nabla f}+\mathcal{T}_g$ |
| $\min_x f(x)+g(x)$ | $\\|\nabla f\\|_2\leq G,\ \nabla^2 f\succeq\mu\cdot I$ | Gradient Descent | $\frac{G^2}{\mu\epsilon}$ | $\mathcal{T}_{\nabla f}+\mathcal{T}_g$ |
| $\min_x f(x)+g(x)$ | $\nabla^2 f\preceq L\cdot I,\ \\|x^*\\|_2\leq R$ | Accelerated Gradient Descent | $\sqrt{\frac{LR^2}{\epsilon}}$ | $\mathcal{T}_{\nabla f}+\mathcal{T}_g$ |
| $\min_x f(x)+g(x)$ | $\mu\cdot I\preceq\nabla^2 f\preceq L\cdot I,$ $\\|x^*\\|_2\leq R$ | Accelerated Gradient Descent | $\sqrt{\frac{L}{\mu}}\log(\frac{LR^2}{\epsilon})$ | $\mathcal{T}_{\nabla f}+\mathcal{T}_g$ |
| $\min_{x\in\Omega} f(x)$ | $\nabla f$ is $L$ Lipschitz in $\\|\cdot\\|$, $\max_{x,y\in\Omega}\\|x-y\\|\leq R$ | Frank Wolfe | $\frac{LR^2}{\epsilon}$ | $\mathcal{T}_{\nabla f}+\mathcal{T}_\Omega$ |
| $\min_x \mathbf{E}_\omega f_\omega(x)$ | $\mathbf{E}_\omega\\|\nabla f_\omega\\|_2^2\leq G^2,$ $\\|x^*\\|_2\leq R$ | Stochastic Gradient Descent | $(\frac{GR}{\epsilon})^2$ | $\mathcal{T}_{\nabla f_\omega}$ |
| $\min_x f(x)\overset{\text{def}}{=}\frac{1}{n}\sum_i f_i(x)$ | $\nabla^2 f\succeq\mu\cdot I,\nabla^2 f_i\preceq L_i\cdot I$ | Accelerated Stochastic Descent | Total Cost: $(\mathcal{T}_{\nabla f}+\sqrt{\frac{1}{\mu}\sum_i L_i}\mathcal{T}_{\nabla f_i})\log(\frac{1}{\epsilon})$ | |
| $\min_{x\in\mathbb{R}^n} f(x)$ | $\nabla^2 f\succeq\mu\cdot I,\partial_i^2 f\leq L_i$ | Accelerated Coordinate Descent | Total Cost: $\sum_i\sqrt{\beta_i/\mu}\mathcal{T}_{\partial_i f}\log(\frac{1}{\epsilon})$ | |

**Table 3.1:** Black Box Problems. The goal is to find $x$ such that $f(x)-\min_x f\leq\epsilon$ with initial point $x^{(0)}=0$. $\mathcal{T}_{\nabla f}$ is the cost of computing $\nabla f(x)$ for any $x$. $\mathcal{T}_g$ is the cost of computing $\arg\min_x g(x)+\frac{\lambda}{2}\\|x-c\\|^2$ for any $c$. $\mathcal{T}_\Omega$ is the cost of computing $\arg\min_{x\in\Omega}c^\top x$ for any $c$. $\mathcal{T}_{\nabla f_\omega}$, $\mathcal{T}_{\nabla f_i}$ and $\mathcal{T}_{\partial_i f}$ are the cost of computing $\nabla f_\omega$, $\nabla f_i$ and $\partial_i f$ for one of the $\omega$ or $i$.

| Convex Problem | Assumption | Algorithm | Depth | Total Work |
|---|---|---|---|---|
| $Ax = b$ | - | Fast Matrix Multiplication | $\log^{O(1)}(n)\log(\frac{1}{\epsilon})$ | $n^{2.3728\cdots}\log(\frac{1}{\epsilon})$ |
| | $\mu \cdot I \preceq A \preceq L \cdot I$ | Conjugate Gradient | $\sqrt{\frac{L}{\mu}}\log(\frac{1}{\epsilon})$ | $\mathrm{nnz}(A)\sqrt{\frac{L}{\mu}}\log(\frac{1}{\epsilon})$ |
| | $A_{ii} \geq \sum_{j\neq i}\lvert A_{ji}\rvert$ | Sparse LU Factorization | $\log^{O(1)}(\frac{nL}{\mu})\log(\frac{1}{\epsilon})$ | $\mathrm{nnz}(A)\log^{O(1)}(\frac{nL}{\mu})\log(\frac{1}{\epsilon})$ |
| $\min_x \lVert Ax - b\rVert_2$ | - | Subspace Embedding / Row Sampling | $\log^{O(1)}(n)\log(\frac{1}{\epsilon})$ | $(\mathrm{nnz}(A) + r^{2.3728\cdots})\log(\frac{1}{\epsilon})$ |
| | Each row of $A$ has $\leq 2$ non-zeros | Sparse Cholesky Factorization | $\log^{O(1)}(n)\log(\frac{1}{\epsilon})$ | $\mathrm{nnz}(A)\log^{O(1)}(n)\log(\frac{1}{\epsilon})$ |
| $\min_{Ax=b, x\in\mathbb{R}^n_+} c^\top x$ | - | Interior Point Method | $\sqrt{n}\log^{O(1)}(n)\log(\frac{1}{\epsilon})$ | $n^{2.3728\cdots}\log(\frac{1}{\epsilon})$ |
| | | Interior Point Method | $\sqrt{r}\log^{O(1)}(n)\log(\frac{1}{\epsilon})$ | $\sqrt{r}(\mathrm{nnz}(A) + r^2)\log^{O(1)} n\log(\frac{1}{\epsilon})$ |
| $\min_x \sum_{i=1}^{n} f_i(a_i^\top x)$ | - | Interior Point Method | $\sqrt{n}\log^{O(1)}(n)\log(\frac{1}{\epsilon})$ | $n^{2.3728\cdots}\log(\frac{1}{\epsilon})$ |

**Table 3.2:** Structured Problems. $r = \mathrm{rank}(A)$. We omit the precise definition of $\epsilon$, but all algorithms depend on $\log(1/\epsilon)$.

# Chapter 4

# Geometrization and Optimization

## 4.1 Mirror Descent

The cutting plane method is well-suited for minimizing non-smooth convex functions with high accuracy. However, its relatively large polynomial time and quadratic space requirements are not favorable for large scale problems. In this section, we discuss how to minimize a non-smooth function with low accuracy.

### Subgradient method

In this section, we consider the constrained non-smooth minimization problem $\min_{x\in\mathcal{D}} f(x)$. Recall how to define the gradient for non-smooth functions:

**Definition 4.1.1.** For any convex function $f$, we define $\partial f(x)$ be the set of vectors $g$ such that

$$f(y) \geq f(x) + g^{\top}(y-x) \text{ for all } y \in \mathbb{R}^n.$$

**Algorithm 9:** `SubgradientMethod`

```
Input: Initial point $x^{(0)} \in \mathbb{R}^n$, step size $h > 0$.
for $k = 0, 1, \cdots, T-2$ do
    Pick any $g^{(k)} \in \partial f(x^{(k)})$.
    $y^{(k+1)} \leftarrow x^{(k)} - h \cdot g^{(k)}$.
    $x^{(k+1)} \leftarrow \pi_{\mathcal{D}}(y^{(k+1)})$ where $\pi_{\mathcal{D}}(y) = arg\min_{x\in\mathcal{D}} \|x-y\|_2$
end
return $\frac{1}{T}\sum_{k=0}^{T-1} x^{(k)}$.
```

To analyze the algorithm, we first need the following Pythagorean theorem. This shows that when we project a point to a convex set, we get closer to any point in the convex set, and how much closer depends on how much we move.

**Lemma 4.1.2** (Pythagorean Theorem). *Given a convex set $\mathcal{D}$ and a point $y$, let $\pi(y) = arg\min_{x\in\mathcal{D}} \|x-y\|_2$. For any $z \in \mathcal{D}$, we have that*

$$\|z-\pi(y)\|_2^2 + \|\pi(y)-y\|^2 \leq \|z-y\|^2.$$

*Proof.* Let $h(t) = \|(\pi(y) + t(z-\pi(y))) - y\|^2$. Since $h(t)$ is minimized at $t = 0$, we have that $h'(0) \geq 0$. Hence, we have

$$h'(0) = (\pi(y)-y)^{\top}(z-\pi(y)) \geq 0.$$

Hence,

$$\begin{aligned}\|z-y\|^2 &= \|z-\pi(y)+\pi(y)-y\|^2\\ &= \|z-\pi(y)\|^2 + 2(z-\pi(y))^{\top}(\pi(y)-y) + \|\pi(y)-y\|^2\\ &\geq \|z-\pi(y)\|^2 + \|\pi(y)-y\|^2.\end{aligned}$$

□

Now, we are ready to analyze the subgradient method. It basically involves tracking $\|x^{(k+1)} - x^*\|_2^2$.

**Theorem 4.1.3.** *Let $f$ be a convex function that is $G$-Lipschitz in $\ell_2$ norm. After $T$ steps, the subgradient method outputs a point $x$ such that*

$$f(x) \le f(x^*) + \frac{\|x^{(0)} - x^*\|_2^2}{2hT} + \frac{h}{2}G^2$$

*where $x^*$ is any point minimizes $f$.*

*Remark.* If the distance $\|x^{(0)} - x^*\|$ and the Lipschitz constant $G$ are known, we can pick $h = \frac{\|x^{(0)}-x^*\|_2}{G\sqrt{T}}$ and get

$$f(x) \le f(x^*) + \frac{G \cdot \|x^{(0)} - x^*\|_2}{\sqrt{T}}.$$

*Proof.* Let $x^*$ be any point minimizes $f$. Then, we have

$$\begin{aligned}\|x^{(k+1)} - x^*\|_2^2 &= \|\pi_{\mathcal{D}}(y^{(k+1)}) - x^*\|_2^2 \\ &\le \|y^{(k+1)} - x^*\|_2^2 \\ &= \|x^{(k)} - hg^{(k)} - x^*\|_2^2 \\ &= \|x^{(k)} - x^*\|_2^2 - 2h\left\langle g^{(k)}, x^{(k)} - x^*\right\rangle + h^2\|g^{(k)}\|_2^2.\end{aligned}$$

where we used Lemma 4.1.2 in the inequality. Using the definition of subgradient, we have that

$$f(x^*) \ge f(x^{(k)}) + \left\langle g^{(k)}, x^* - x^{(k)}\right\rangle.$$

Therefore, we have that

$$\begin{aligned}\|x^{(k+1)} - x^*\|_2^2 &\le \|x^{(k)} - x^*\|_2^2 - 2h\cdot(f(x^{(k)}) - f(x^*)) + h^2\|g^{(k)}\|_2^2 \\ &\le \|x^{(k)} - x^*\|_2^2 - 2h\cdot(f(x^{(k)}) - f(x^*)) + h^2G^2.\end{aligned}$$

Note that this equation shows that if the error $f(x^{(k)}) - f(x^*)$ is large, then we move closer to optimum. Rearrange the terms, we have

$$f(x^{(k)}) - f(x^*) \le \frac{1}{2h}\left(\|x^{(k)} - x^*\|_2^2 - \|x^{(k+1)} - x^*\|_2^2\right) + \frac{h}{2}G^2.$$

Now, we sum over all iterations, we have

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1}(f(x^{(k)}) - f(x^*)) &\le \frac{1}{T}\cdot\frac{1}{2h}(\|x^{(0)} - x^*\|_2^2 - \|x^{(T)} - x^*\|_2^2) + \frac{h}{2}G^2 \\ &\le \frac{\|x^{(0)} - x^*\|_2^2}{2hT} + \frac{h}{2}G^2.\end{aligned}$$

The result follows from $f(\frac{1}{T}\sum_{k=0}^{T-1}x^{(k)}) \le \frac{1}{T}\sum_{k=0}^{T-1}(f(x^{(k)}) - f(x^*))$. □

## Intuition of Mirror Descent

Consider using subgradient method to minimize $f(x) = \sum_i |x|$ over the unit ball $B(0,1)$. The subgradient of $f$ is given by $\text{sign}(x)$ (assuming $x_i \neq 0$ for all $i$). Therefore, we have $G = \sqrt{n}$ and $R = 1$. Hence, the convergence rate is $\sqrt{\frac{n}{k}}$ which is not dimension independent. Intuitively, the dimension dependence comes from the fact we must take a tiny step size to avoid affecting all the variables. For example, if we take a constant step size $h$ and start at the point $x = (1, \frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n})$, then we will get a point with $x_i$ is constant in all direction $i \neq 1$ and this increases the function $f$ dramatically from $\Theta(1)$ to $\Theta(nh)$. Therefore, to get a constant error, we need to take step size $h = \frac{1}{n}$ and hence we need $n$ iterations.

Conceptually, the sentence $x = x - \eta g$ does not make sense either. Imagine now the problem is infinitely dimensional. Note that $f(x) < +\infty$ for any $x \in \ell_1$. (Recall that $\ell_1 = \{x \in \mathbb{R}^{\mathbb{N}} : \sum |x_i| < \infty\}$.) On the other hand, its gradient $g$ lives in $\ell_\infty$ space. Since $x$ and $g$ are not in the same space, the term $x - \eta g$ does not make sense. More precisely, we have the following tautology (directly follow from the definition of dual space).

**Lemma 4.1.4.** *Given a Banach space $\mathcal{D}$ and a continuously differentiable function $f$ from $\mathcal{D}$ to $\mathbb{R}$. Then, we have that $\nabla f(x) \in \mathcal{D}^*$ for any $x$.*

In general, if the function $f$ is on the primal space $\mathcal{D}$, then the gradient $g$ lives in the dual space $\mathcal{D}^*$. Therefore, we need to map $x$ from the primal space $\mathcal{D}$ to the dual space $\mathcal{D}^*$, update its position, then map the point back to the original space $\mathcal{D}$.

In fact, Lemma 4.1.4 gives us one such map, $\nabla f$. Consider the following algorithm: Starting at $x$. We use $\nabla f(x)$ to map $x$ to the dual space $y = \nabla f(x)$. Then, we apply the gradient step on the dual $y^{(\text{new})} = y - \nabla f(x)$ and map it back to the primal space, namely finding $x^{(\text{new})}$ such that $\nabla f(x^{(\text{new})}) = y^{(\text{new})}$. Note that $y^{(\text{new})} = 0$ and hence $x^{(\text{new})}$ is exactly the minimizer of $f$. So, if we can map it back, this algorithm solves the problem in one step. Unfortunately, the task of mapping it back is exactly our original problem.

Instead of using the same $f$, the mirror descent uses some other function $\Phi$, called the mirror map. For constrained problems, the mirror map may not bring the point back to a point in $\mathcal{D}$. Naively, one may consider the algorithm

$$\begin{aligned}\nabla\Phi(y^{(t+1)}) &= \nabla\Phi(x^{(t)}) - h\cdot\nabla f(x^{(t)}),\\ x^{(t+1)} &= \arg\min_{x\in\mathcal{D}} \|x - y^{(t+1)}\|_2.\end{aligned}$$

Note that the first step of finding $y^{(t+1)}$ involves solving an optimization problem. We will show how to do this (See 4.1.1) but not for this version. This algorithm is bad because it does not consider the geometry of the function $\Phi$. Instead we should measure the distance according to $\Phi$:

**Definition 4.1.5.** For any strictly convex function $\Phi$, we define the Bregman divergence as

$$D_\Phi(y,x) = \Phi(y) - \Phi(x) - \langle\nabla\Phi(x), y - x\rangle .$$

Note that $D_\Phi(y,x)$ is the error of the first order Taylor expansion of $\Phi$ at $x$. Due to the convexity of $\Phi$, we have that $D_\Phi(y,x) \geq 0$. Also, we note that $D_\Phi(y,x)$ is convex in $y$, but not necessarily in $x$.

**Example 4.1.6.** $D_\Phi(y,x) = \|y-x\|^2$ for $\Phi(x) = \|x\|^2$. $D_\Phi(y,x) = \sum_i y_i \log\frac{y_i}{x_i} - \sum y_i + \sum x_i$ for $\Phi(x) = \sum x_i \log x_i$.

**Algorithm 10:** `MirrorDescent`

```
Input: Initial point $x^{(0)} = \arg\min_{x\in\mathcal{D}} \Phi(x)$, step size $h > 0$.
for $k = 0, 1, \cdots, T-2$ do
    Pick any $g^{(k)} \in \partial f(x^{(k)})$.
    // As shown in (4.1.1), the next 2 steps can be implemented by an optimization over $D_\Phi$.
    Find $y^{(k+1)}$ such that $\nabla\Phi(y^{(k+1)}) = \nabla\Phi(x^{(k)}) - h\cdot g^{(k)}$.
    $x^{(k+1)} \in \pi_{\mathcal{D}}^{\Phi}(y^{(k+1)})$ where $\pi_{\mathcal{D}}^{\Phi}(y) = \arg\min_{x\in\mathcal{D}} D_\Phi(x,y)$.
end
return $\frac{1}{T}\sum_{k=0}^{T-1} x^{(k)}$.
```

Mirror descent step can be written as follows, where in the second step we use the fact that the optimization is only over $x$:

$$\begin{aligned}x^{(k+1)} &= \arg\min_{x\in\mathcal{D}} D_\Phi(x, y^{(k+1)})\\ &= \arg\min_{x\in\mathcal{D}} \Phi(x) - \Phi(y^{(k+1)}) - \nabla\Phi(y^{(k+1)})^\top(x - y^{(k+1)})\\ &= \arg\min_{x\in\mathcal{D}} \Phi(x) - \nabla\Phi(y^{(k+1)})^\top x\\ &= \arg\min_{x\in\mathcal{D}} \Phi(x) - \nabla\Phi(x^{(k)})^\top x + hg^{(k)\top} x\\ &= \arg\min_{x\in\mathcal{D}} hg^{(k)\top} x + D_\Phi(x, x^{(k)}). \qquad (4.1.1)\end{aligned}$$

Note that this is a natural generalization of $x^{(k+1)} = \arg\min_{x\in\mathcal{D}} hg^{(k)\top}x + \|x - x^{(k)}\|^2$.

## Analysis of Mirror Descent

**Lemma 4.1.7** (Pythagorean Theorem)**.** *Given a convex set $\mathcal{D}$ and a point $y$. Let $\pi(y) = \arg\min D_\Phi(x, y)$. For any $z \in \mathcal{D}$, we have that*

$$D_\Phi(z, \pi(y)) + D_\Phi(\pi(y), y) \le D_\Phi(z, y).$$

*Proof.* Let $h(t) = D_\Phi(\pi(y) + t(z - \pi(y)), y)$. Since $h(t)$ is minimized at $t = 0$, we have that $h'(0) \ge 0$. Hence, we have

$$h'(0) = (\nabla\Phi(\pi(y)) - \nabla\Phi(y))^\top (z - \pi(y)) \ge 0.$$

Hence,

$$\begin{aligned} D_\Phi(z, y) &= D_\Phi(z, \pi(y)) + 2(z - \pi(y))^\top (\nabla\Phi(\pi(y)) - \nabla\Phi(y)) + D_\Phi(\pi(y), y) \\ &\ge D_\Phi(z, \pi(y)) + D_\Phi(\pi(y), y). \end{aligned}$$

□

**Theorem 4.1.8.** *Let $f$ be a $G$-Lipschitz convex function on some norm space $\|\cdot\|$. Let $\Phi$ be a $\rho$-strongly convex function on $\mathcal{D}$ with respect to $\|\cdot\|$ with diameter square $R^2 = \sup_{x\in\mathcal{D}} \Phi(x) - \Phi(x^{(0)})$. Then, the mirror descent outputs $x$ such that*

$$f(x) - \min_x f(x) \le \frac{R^2}{hT} + \frac{h}{2\rho} G^2.$$

*Remark* 4.1.9. We call a function $f$ is $\rho$ strongly convex w.r.t $\|\cdot\|$ if for any $x, y$, we have

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

*Remark* 4.1.10. Picking $h = \frac{R}{G}\sqrt{\frac{2\rho}{T}}$, we get the rate $f(x) \le \min_x f(x) + GR\sqrt{\frac{2}{\rho T}}$.

*Proof.* This proof is a complete "mirror" of the proof in Theorem 4.1.3.

$$\begin{aligned} D_\Phi(x^*, x^{(k+1)}) &\le D_\Phi(x^*, y^{(k+1)}) - D_\Phi(x^{(k+1)}, y^{(k+1)}) \\ &= D_\Phi(x^*, x^{(k)}) + (x^* - x^{(k)})^\top (\nabla\Phi(x^{(k)}) - \nabla\Phi(y^{(k+1)})) + D_\Phi(x^{(k)}, y^{(k+1)}) - D_\Phi(x^{(k+1)}, y^{(k+1)}) \\ &= D_\Phi(x^*, x^{(k)}) - h \cdot \left\langle g^{(k)}, x^{(k)} - x^* \right\rangle + D_\Phi(x^{(k)}, y^{(k+1)}) - D_\Phi(x^{(k+1)}, y^{(k+1)}) \end{aligned}$$

where we used Lemma 4.1.7 in the inequality. Using the definition of subgradient, we have that

$$f(x^*) \ge f(x^{(k)}) + \left\langle g^{(k)}, x^* - x^{(k)} \right\rangle.$$

Therefore, we have that

$$D_\Phi(x^*, x^{(k+1)}) \le D_\Phi(x^*, x^{(k)}) - h \cdot (f(x^{(k)}) - f(x^*)) + D_\Phi(x^{(k)}, y^{(k+1)}) - D_\Phi(x^{(k+1)}, y^{(k+1)}).$$

Next we note that

$$\begin{aligned} & D_\Phi(x^{(k)}, y^{(k+1)}) - D_\Phi(x^{(k+1)}, y^{(k+1)}) \\ =& \Phi(x^{(k)}) - \Phi(x^{(k+1)}) - \nabla\Phi(y^{(k+1)})^\top (x^{(k)} - x^{(k+1)}) \\ \le& \left\langle \nabla\Phi(x^{(k)}) - \nabla\Phi(y^{(k+1)}), x^{(k)} - x^{(k+1)} \right\rangle - \frac{\rho}{2}\|x^{(k)} - x^{(k+1)}\|^2 \\ =& h \cdot \left\langle g^{(k)}, x^{(k)} - x^{(k+1)} \right\rangle - \frac{\rho}{2}\|x^{(k)} - x^{(k+1)}\|^2 \\ \le& hG\|x^{(k)} - x^{(k+1)}\| - \frac{\rho}{2}\|x^{(k)} - x^{(k+1)}\|^2 \\ \le& \frac{(hG)^2}{2\rho}. \end{aligned}$$

Hence, we have

$$D_\Phi(x^*, x^{(k+1)}) \le D_\Phi(x^*, x^{(k)}) - h \cdot (f(x^{(k)}) - f(x^*)) + \frac{(hG)^2}{2\rho}.$$

Note that this equation shows that if the error $f(x^{(k)})-f(x^*)$ is large, then we move closer to optimum. Rearranging the terms, we have

$$f(x^{(k)})-f(x^*)\leq\frac{1}{h}\left(D_\Phi(x^*,x^{(k)})-D_\Phi(x^*,x^{(k+1)})\right)+\frac{hG^2}{2\rho}.$$

Summing over all iterations, we have

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1}(f(x^{(k)})-f(x^*))&\leq\frac{1}{T}\cdot\frac{1}{h}(D_\Phi(x^*,x^{(0)})-D_\Phi(x^*,x^{(T)}))+\frac{h}{2\rho}G^2\\&\leq\frac{R^2}{hT}+\frac{h}{2\rho}G^2.\end{aligned}$$

The result follows from $f(\frac{1}{T}\sum_{k=0}^{T-1}x^{(k)})\leq\frac{1}{T}\sum_{k=0}^{T-1}(f(x^{(k)})-f(x^*))$. $\square$

## Multiplicative Weight Update

In this section, we discuss a bit the mirror descent under the map $\Phi(x)=\sum x_i\log x_i$ and the simplex $\mathcal{D}=\{x_i\geq 0,\sum x_i=1\}$.

**Step Formula** As we showed in (4.1.1), we have that

$$x^{(k+1)}=\arg\min_{x\in\mathcal{D}}hg^{(k)\top}x+D_\Phi(x,x^{(k)}).$$

Note that

$$\begin{aligned}D_\Phi(x,x^{(k)})&=\sum x_i\log x_i-\sum x_i^{(k)}\log x_i^{(k)}-\sum_i(1+\log x_i^{(k)})(x_i-x_i^{(k)})\\&=\sum x_i\log\frac{x_i}{x_i^{(k)}}\end{aligned}$$

where we used that $\sum_i x_i=\sum_i x_i^{(k)}$. Hence, the step is simply

$$x^{(k+1)}=\arg\min_{\sum x_i=1,x_i\geq 0}hg^{(k)\top}x+\sum x_i\log\frac{x_i}{x_i^{(k)}}.$$

Note that the optimality condition is given by

$$hg_i^{(k)}+\log\frac{x_i^{(k+1)}}{x_i^{(k)}}+1=\lambda$$

for some Lagrangian multiplier $\lambda$. Rewriting, we have

$$x_i^{(k+1)}=e^{-hg_i^{(k)}}x_i^{(k)}/Z$$

for some normalization constant $Z$. Note that this algorithm multiplies the current $x$ with a multiplicative factor and hence it is also called multiplicative weight update.

**Strong Convexity** To bound the strong convexity, we note that

$$\Phi(y)-\Phi(x)-\langle\nabla\Phi(x),y-x\rangle=\frac{1}{2}(y-x)^\top\nabla^2\Phi(\zeta)(y-x)$$

for some $\zeta$ between $x$ and $y$. Since $\nabla^2\Phi(\zeta)=\frac{1}{\zeta}$, we have

$$(y-x)^\top\nabla^2\Phi(\zeta)(y-x)=\sum\frac{(y_i-x_i)^2}{\zeta_i}\geq\frac{(\sum_i|y_i-x_i|)^2}{\sum_i\zeta_i}=(\sum_i|y_i-x_i|)^2$$

where we used that $\sum_i x_i=\sum_i y_i=1$ and so is $\sum_i\zeta_i$. Hence, we have that

$$\Phi(y)-\Phi(x)-\langle\nabla\Phi(x),y-x\rangle\geq\frac{1}{2}\|y-x\|_1^2.$$

Therefore, $\Phi$ is 1 strongly convex in $\|\cdot\|_1$. Hence, $\rho=1$.

**Diameter** Direct calculations shows that $-\log n \leq \Phi(x) \leq 0$. Hence, $R^2 = \log n$.

**Result**

**Theorem 4.1.11.** *Let $f$ be 1-Lipschitz functions on $\|\cdot\|_1$. Then, the mirror descent with the mirror map discussed above outputs $x$ in $T$ iterations such that*

$$f(x) - \min_x f(x) \leq \sqrt{\frac{2\log n}{T}}.$$

In comparison, projected gradient descent gives $\sqrt{\frac{n}{T}}$ which is much larger.

## Online Guarantee

We note that the proof of Theorem 4.1.3 and Theorem 4.1.8 does not use the fact that we are solving the same convex function in each step, except for the very last step. Suppose at the $k$-th step, the function we compute the gradient is $f^{(k)}$. Then, we proved that

$$\frac{1}{T}\sum_{k=0}^{T-1}(f^{(k)}(x^{(k)}) - f^{(k)}(x)) \leq GR\sqrt{\frac{2}{\rho T}} \tag{4.1.2}$$

for any $k$. Also, we note that the point $x^{(k)}$ depends on $f^{(1)}, f^{(2)}, \cdots, f^{(k-1)}$ but not $f^{(k)}$. Therefore, we can view the whole progress as a game:

- The player chooses a point $x^{(k)}$.
- The adversary reveals $\nabla f^{(k)}(x^{(k)})$.
- The player gets the loss $f^{(k)}(x^{(k)})$.

In general, we call $\sum_{k=0}^{T-1}(f^{(k)}(x^{(k)}) - f^{(k)}(x))$ is the regret of not choosing the best point $x$. The equation (4.1.2) shows that mirror descent gives a strategy with regret $GR\sqrt{\frac{2T}{\rho}}$. If the domain is the unit ball, all functions are 1-Lipschitz in $\ell_2$ norm, this shows that a regret bound $O(\sqrt{T})$.

**Open Problem 4.1.12.** If the domain is the unit ball, all functions are 1-Lipschitz in $\ell_2$ norm and if the adversary reveals only $f^{(k)}(x^{(k)})$, what is the best possible regret bound? The current best bound [10] is $O(n^c\sqrt{T})$ for some pretty large constant $c$.

## ■ 4.2 Frank–Wolfe Algorithm

Mirror descent is not suitable for all spaces. The guarantee of mirror descent crucially depends on the fact that there is a 1-strongly convex mirror map $\Phi$ such that $\max_x \Phi(x) - \min_x \Phi(x)$ is small on the domain. For some domains such as $\{x : \|x\|_\infty \leq 1\}$, the range $\max_x \Phi(x) - \min_x \Phi(x)$ can be large.

**Lemma 4.2.1.** *Let $\Phi$ be a 1-strongly convex function on $\mathbb{R}^n$ over $\|\cdot\|_\infty \leq 1$. Then,*

$$\max_{\|x\|_\infty \leq 1} \Phi(x) \geq \min_{\|x\|_\infty \leq 1} \Phi(x) + \frac{n}{2}.$$

*Proof.* By the strong convexity of $\Phi$, we have that

$$\begin{aligned}\mathbf{E}_{s_k\in\{\pm1\}\ \forall k\in[n]}\Phi(s_1,\cdots,s_n) &\ge \mathbf{E}_{s_k\in\{\pm1\}\ \forall k\in[n-1]}\Phi(s_1,\cdots,s_{n-1},0)+\langle\nabla\Phi(s_1,\cdots,s_{n-1},0),s_n\rangle+\frac{1}{2}\\ &= \mathbf{E}_{s_k\in\{\pm1\}\ \forall k\in[n-1]}\Phi(s_1,\cdots,s_{n-1},0)+\frac{1}{2}\\ &\vdots\\ &\ge \Phi(0)+\frac{n}{2}.\end{aligned}$$

□

*Remark.* This inequality is tight because $\frac{1}{2}\|x\|^2$ is 1-strongly convex in $\|\cdot\|_\infty$ and its value is between 0 and $\frac{n}{2}$.

## Algorithm

Now, we give another geometry dependent algorithm that relies on a different set of assumptions. The problem we study in this section is of the form

$$\min_{x\in\mathcal{D}} f(x)$$

for $f$ such that $\nabla f$ is "Lipschitz" in a certain sense. The algorithm reduces the problem to a sequence of linear optimization problems.

**Algorithm 11:** `FrankWolfe`

```
Input: Initial point $x^{(0)}\in\mathbb{R}^n$, step size $h>0$.
for $k=0,1,\cdots,T-1$ do
    Compute $y^{(k)}=\arg\min_{y\in\mathcal{D}}\langle y,\nabla f(x^{(k)})\rangle$.
    $x^{(k+1)}\leftarrow(1-h_k)x^{(k)}+h_k y^{(k)}$ with $h_k=\frac{2}{k+2}$.
end
return $x^{(T)}$.
```

## Analysis

**Theorem 4.2.2.** *Given a convex function $f$ on a convex set $\mathcal{D}$ and a constant $C_f$ such that*

$$f((1-h)x+hy))\le f(x)+h\langle\nabla f(x),y-x\rangle+\frac{1}{2}C_f h^2$$

*for any $x,y\in\mathcal{D}$ and $h\in[0,1]$, we have*

$$f(x^{(k)})-f(x^*)\le\frac{2C_f}{k+2}.$$

*Proof.* By the definition of $C_f$, we have that

$$f(x^{(k+1)})\le f(x^{(k)})+h_k\left\langle\nabla f(x^{(k)}),y^{(k)}-x^{(k)}\right\rangle+\frac{1}{2}C_f h_k^2.$$

Note that

$$f(x^*)\ge f(x^{(k)})+\left\langle\nabla f(x^{(k)}),x^*-x^{(k)}\right\rangle\ge f(x^{(k)})+\left\langle\nabla f(x^{(k)}),y^{(k)}-x^{(k)}\right\rangle$$

where we used the fact that $y^{(k)}=\arg\min_{y\in\mathcal{D}}\langle y,\nabla f(x^{(k)})\rangle$ and that $x^*\in\mathcal{D}$. Hence, we have that

$$f(x^{(k+1)})\le f(x^{(k)})-h_k(f(x^{(k)})-f(x^*))+\frac{1}{2}C_f h_k^2.$$

Let $e_k=f(x^{(k)})-f(x^*)$. Then,

$$e_{k+1}\le(1-h_k)e_k+\frac{1}{2}C_f h_k^2.$$

Note that $e_0=f(x^{(0)})-f^*\le\frac{1}{2}C_f$. By induction, we have that $e_k\le\frac{2C_f}{k+2}$. □

*Remark.* Note that this proof is in fact same as Theorem 1.5.5. Also, if $\nabla f$ has $L$ Lipschitz with respect to $\|\cdot\|$ over the domain $\mathcal{D}$, then $C_f \leq L \cdot \mathrm{diam}_{\|\cdot\|}(\mathcal{D})^2$.

## Sparsity Analysis

For the domain $\mathcal{D}$ is simplex, the step $y^{(k)}$ always has a single non-zero. Hence, each step of Frank-Wolfe increases the sparsity by at most 1. So, we can think that the convergence result of Frank Wolfe proves that there is an approximate sparse solution of the optimization problem. In particular, this shows that Frank Wolfe is optimal in the following sense.

**Lemma 4.2.3.** *Let $f(x) = \|x\|_2^2$ and $\mathcal{D} = \{\sum_{i=1}^n x_i = 1, x_i \geq 0\}$. Then $\min_{x \in \mathcal{D}, \|x\|_0 \leq k} f(x) = \frac{1}{k}$.*

*Proof.* The minimizer is given by setting $k$ of the coordinates of $x$ to $\frac{1}{k}$ each and 0 otherwise. □

More generally, if $\mathcal{D} = \mathrm{conv}(z_i)$, then the step $y^{(k)}$ is always given by one of the point in $z_i$. Therefore, the solution is always given by a small combination of vertices $z_i$. In particular, it shows that any point in a convex set can be approximated by a small combination of vertices:

**Lemma 4.2.4.** *Let polytope $P = \mathrm{conv}(v_i)$ be a polytope that lies inside an unit ball. For any $u \in P$ and for any $\epsilon > 0$, there are $k = O(\frac{1}{\epsilon^2})$ vertices $v_1, \cdots, v_k$ of $P$ such that*

$$\|\sum_{i=1}^k \lambda_i v_i - u\|_2 \leq \epsilon$$

*for some $\sum_i \lambda_i = 1$, $\lambda_i \geq 0$.*

*Remark.* Using mirror descent instead, one can prove there are vertices with $\lambda_i = \frac{1}{k}$ for all $i$ [47].

*Proof.* We run Frank-Wolfe on $P$ with the function $f(x) = \frac{1}{2}\|x - u\|_2^2$. Note that $f$ is 1 Lipschitz with respect to $\|x\|_2$ and that the diameter of $P$ is bounded by 1. Hence, we have that

$$C_f \leq 1.$$

Therefore, Frank-Wolfe shows that

$$f(x^{(k)}) - f(u) \leq \frac{2}{k+2}.$$

Since $f(u) = 0$, we have that $\|x^{(k)} - u\|^2 \leq \frac{4}{k+2}$. This gives the result. □

Here, let us give another proof. For any $x \in P$, we can write $x = \sum_i c_i v_i$ for $c_i$ such that $\sum_i c_i = 1$, $c_i \geq 0$. We can view $c$ as a probability. Let $X$ be a random point constructed by sampling $v_i$ with probability $c_i$. Let $X_T$ be the average of $T$ independent copies of $X$. Then, we have that

$$\mathbf{E}\|X_T - x\|^2 = \frac{1}{T}\mathbf{E}\|X - x\|^2 \leq \frac{\mathbf{E}\|X\|^2}{T} = \frac{1}{T}.$$

Therefore, any point in $P$ can be approximated by combinations of $T$ points with error at most $\frac{1}{\sqrt{T}}$. Note that this gives a tighter bound than Frank-wolfe, but one need to solve a linear program to find the coefficients $c$.

## Applications to Maximum Flow* (in progress)

In this section, we discuss how to approximate the maximum flow in an undirected graph in almost linear time. The key fact we use about flow is the following.

**Theorem 4.2.5** (Oblivious routing [46, 54, 29, 51])**.** *For any graph $G = (V, E)$, there is a map $R : \mathbb{R}^{|V|} \to \mathbb{R}^{|E|}$, called oblivious routing, such that for any demand $d \in \mathbb{R}^{|V|}$ with $\sum_{v \in V} d_V = 0$, we have*

1. *$B^\top Rd = d$ (the routing meet the demand $d$),*
2. *$\|Rd\|_\infty \leq n^{o(1)} \cdot \min_{B^\top f=d} \|f\|_\infty$ (we call $n^{o(1)}$ is the competitive ratio; it represents how good this routing compared to the optimum).*

*Moreover, $Rv$ and $R^\top v$ can be computed in $n^{1+o(1)}$ time.*

**Open Problem 4.2.6.** Do we have an oblivious routing that can be computed in $n \log^{O(1)} n$ time with the competitive ratio $\log^{O(1)} n$?

First, we note that the maximum flow problem is same as finding a unit flow from $s$ to $t$ with minimum $\|f\|_\infty$, namely, finding

$$\min_{B^\top f=1_{st}} \|f\|_\infty$$

where $B : \mathbb{R}^{|V|} \to \mathbb{R}^{|E|}$ is the incidence matrix of the graph and $1_{st}$ is the vector with $+1$ at $s$ and $-1$ at $t$. Let $v$ be the value of the maximum flow. Then the above problem has minimum value $\frac{1}{v}$. Let $f^{(0)} = R1_{st}$. Then, we know that $\|f^{(0)}\|_\infty \leq \frac{n^{o(1)}}{v}$ because the oblivious routing is as good as the maximum flow up to $n^{o(1)}$ factor.

Now, we can further simplify the problem by

$$\min_{B^\top f=0} \left\| f + f^{(0)} \right\|_\infty .$$

To remove the constraint, we note the following:

**Lemma 4.2.7.** *Let $R$ be an oblivious routing. Then $\{f : B^\top f = 0\} = \{Pf : f \in \mathbb{R}^{|E|}\}$ where $P = I - RB^\top$. Furthermore, we have that $\|P\|_{\infty\to\infty} = n^{o(1)}$.*

*Proof.* The set $\{f : B^\top f = 0\}$ is simply the set of circulations. Note that $B^\top Pf = B^\top - B^\top RB^\top = B^\top - B^\top = 0$. Hence, all $Pf$ are circulations.

On the other any, given any circulation $f$, then, $Pf = f - RB^\top f = f$. Hence, any circulation $f$ is given by $Pf$.

For the last result, we note that $\|Pf\|_{\infty\to\infty} \leq 1 + \|RB^\top\|_{\infty\to\infty}$. For the last term, note that for any $g$, $\|RB^\top g\|_\infty \leq n^{o(1)} \min_{B^\top f=B^\top g} \|f\|_\infty \leq n^{o(1)}\|g\|_\infty$. Hence, we have $\|RB^\top\|_{\infty\to\infty} \leq n^{o(1)}$. □

Due to the lemma above, we have that

$$\min_{B^\top f=0} \left\| f + f^{(0)} \right\|_\infty = \min_f \left\| Pf + f^{(0)} \right\|_\infty .$$

Now, we can apply Frank-Wolfe on the right problem.

**Theorem 4.2.8** ([54, 29])**.** *We can find a flow with value at least $(1-\varepsilon)v$ in time*

$$O^*(\frac{m}{\varepsilon^2}).$$

*Remark.* [55] designed a specified first-order method for $\|\cdot\|_\infty$ and obtained a running time $O(m \log^{O(1)} n/\epsilon)$.

*Proof.* To smoothen $\|\cdot\|_\infty$, we consider the potential

$$\ell_\delta(f) = \text{smax}_\delta(Pf + f^{(0)})$$

where

$$\text{smax}_\delta(f) = \delta \log(\sum_i e^{f_i/\delta} + e^{-f_i/\delta}).$$

Suppose that $\ell_\delta(f) \leq \min_f \ell_\delta(f) + \frac{\varepsilon}{2v}$. Then, we have that

$$\left\| Pf + f^{(0)} \right\|_\infty \leq \ell_\delta(t) \leq \min_f \ell_\delta(f) + \frac{\varepsilon}{2v} \leq \min_f \ell(f) + \frac{\varepsilon}{2v} + \delta \log(2m).$$

Setting $\delta = \frac{\varepsilon}{2v\log(2m)}$, we have that $\left\|Pf + f^{(0)}\right\|_\infty \leq \min_f \ell(f) + \frac{\varepsilon}{v}$. Since $B^\top(Pf + f^{(0)}) = B^\top f^{(0)} = 1_{st}$, $Pf + f^{(0)}$ satisfies the requirements.

Now, we analyze how fast Frank-Wolfe takes to solve the problem. It is easy to prove that $\nabla \ell_\delta$ has Lipschitz constant in $\ell_\infty$ bounded by

$$O(\frac{\|P\|_{\infty\to\infty}^2}{\delta}) = O(\frac{v}{\varepsilon} n^{o(1)}).$$

Let $f^*$ be the minimizer of $\ell_\delta$. Then, $Pf^*$ is also a minimizer and

$$\|Pf^*\|_\infty \leq \left\|Pf^* + f^{(0)}\right\|_\infty + \left\|f^{(0)}\right\|_\infty = O^*(\frac{1}{v}).$$

Therefore, the diameter $R = O^*(\frac{1}{v})$. Therefore, the Frank-wolfe shows that

$$\ell_\delta(f^{(k)}) \leq \ell_\delta(f^*) + \frac{O(\frac{v}{\varepsilon} n^{o(1)}) \cdot O^*(\frac{1}{v^2})}{k} \leq \ell_\delta(f^*) + \frac{n^{o(1)}}{v\epsilon k}.$$

Hence, after $k = n^{o(1)}/\epsilon^2$ iterations, we find the approximate maximum flow. □

## 4.3 Newton Method

We begin with the Newton-Raphson method for finding the zeros of a function $g$, i.e. finding $x$ such that $g(x) = 0$. In optimization, the goal is to find a root of the gradient, $\nabla f(x) = 0$. The Newton-Raphson iteration approximates a function by its gradient/Jacobian, then guesses where the gradient intersects the zero line as a likely zero, repeating this process until a sufficient approximation has been found. In one dimension, we approximate the function $g$ by its gradient

$$g(x) \sim g(x^{(k)}) + g'(x^{(k)})(x - x^{(k)}).$$

Finding the zeros of the right hand side, we have the Newton step

$$x^{(k+1)} = x^{(k)} - \frac{g(x^{(k)})}{g'(x^{(k)})}.$$

In high dimension, we can approximate the function by its Jacobian $g(x) \sim g(x^{(k)}) + Dg(x^{(k)})(x - x^{(k)})$ and this gives the step

$$x^{(k+1)} = x^{(k)} - \left(Dg(x^{(k)})\right)^{-1} g(x^{(k)}).$$

When the function $g(x) = \nabla f(x)$, then the Newton step becomes

$$x^{(k+1)} = x^{(k)} - \left(\nabla^2 f(x^{(k)})\right)^{-1} \nabla f(x^{(k)}).$$

Alternatively, we can derive the Newton step by $x^{(k+1)} \leftarrow \min_x f(x^{(k)}) + \langle \nabla f(x^{(k)}), x - x^{(k)} \rangle + \frac{1}{2}(y - x^{(k)})^\intercal (\nabla^2 f(x^{(k)}))(y - x^{(k)})$, namely Newton step finds the minimum of the second order approximation of the function. We note that Newton's iteration is a fairly natural idea, since

1. It is affine invariant (changing coordinate systems won't affect the convergence).

2. Newton step is the fastest descent direction taking the Hessian into account.

For nice enough functions, gradient methods converges to the solution linearly (namely, it takes $c \cdot \log \frac{1}{\epsilon}$ iterations for some $c$ depending on the problem) while the Newton method converges to the solution quadratically (namely, it takes $c' \cdot \log\log\frac{1}{\epsilon}$ for some $c'$). However, each step of Newton method involves solving a linear system, which can be much more expensive.

**Algorithm 12:** `NewtonMethod`

```
Input: Initial point $x^{(0)} \in \mathbb{R}^n$
for $k = 0, 1, \cdots, T-1$ do
    $x^{(k+1)} = x^{(k)} - \left(Dg(x^{(k)})\right)^{-1} g(x^{(k)})$.
end
return $x^{(T)}$.
```

**Theorem 4.3.1** (Quadratic convergence)**.** *Assume that $g : \mathbb{R}^n \to \mathbb{R}^n$ is twice continuously differentiable. Let $x^{(k)}$ be the sequence given by the Newton method. Suppose that $x^{(k)}$ converges to some $x^*$ such that $g(x^*) = 0$ and $Dg(x^*)$ is invertible. Then, for $k$ large enough, we have*

$$\|x^{(k+1)} - x^*\| \le \|(Dg(x^*))^{-1} D^2 g(x^*)\|_{\mathrm{op}} \cdot \|x^{(k)} - x^*\|^2.$$

*Proof.* Let $e^{(k)} = x^* - x^{(k)}$. Then, we have that

$$g(x^*) = g(x^{(k)}) + Dg(x^{(k)})[e^{(k)}] + \int_0^1 (1-s) D^2 g(x^{(k)} + s e^{(k)})[e^{(k)}, e^{(k)}] ds.$$

Hence, we have

$$0 = Dg(x^{(k)})^{-1} g(x^{(k)}) + x^* - x^{(k)} + \int_0^1 (1-s) Dg(x^{(k)})^{-1} D^2 g(x^{(k)} + s e^{(k)})[e^{(k)}, e^{(k)}] ds.$$

Since $x^{(k+1)} = x^{(k)} - Dg(x^{(k)})^{-1} g(x^{(k)})$, we have

$$x^{(k+1)} - x^* = \int_0^1 (1-s) Dg(x^{(k)})^{-1} D^2 g(x^{(k)} + s e^{(k)})[e^{(k)}, e^{(k)}] ds.$$

So, we have

$$\begin{aligned}\|x^{(k+1)} - x^*\| &\le \int_0^1 (1-s) \|Dg(x^{(k)})^{-1} D^2 g(x^{(k)} + s e^{(k)})[e^{(k)}, e^{(k)}]\| ds \\ &\le \int_0^1 (1-s) \|Dg(x^{(k)})^{-1} D^2 g(x^{(k)} + s e^{(k)})\|_{\mathrm{op}} ds \cdot \|x^{(k)} - x^*\|^2.\end{aligned}$$

Since $x^{(k)} \to x^*$, we have that $Dg(x^{(k)})^{-1} D^2 g(x^{(k)} + s e^{(k)}) \to Dg(x^*)^{-1} D^2 g(x^*)$ and hence the result. □

This argument above only uses that $g \in \mathcal{C}^2$ and does not require convexity. Without further global assumptions on $g$, Newton method does not always converge to the root. The argument above only shows that if the algorithm converges, then it converges quadratically eventually. We call such analysis "local" convergence since it only gives a bound when $x^{(k)}$ is close enough to $x^*$. In comparison, all the analysis before in this book are global convergence, bounding the total number of iterations. In practice, both analysis are important, global convergence makes sure the algorithm is robust and local quadratic convergence makes sure the algorithm converges to machine accuracy fast enough. The local quadratic convergence is particular important for simple problems such as computing $\sqrt{x}$.

### Finding the largest roots of polynomials

In general, Newton method does not converges to the root globally. Here, we give an application that does converges "globally".

**Theorem 4.3.2.** *Given a polynomial $g(x) = \sum_{i=1}^n a_i x^i$ with only real roots. Let $\lambda_1$ be its largest root. Suppose that $\lambda_1 \le x^{(0)}$. Then, after $O(n \log(\frac{1}{\epsilon}))$ iterations, the Newton method finds $x^{(k)}$ such that*

$$\lambda_1 \le x^{(k)} \le \lambda_1 + \varepsilon \cdot (x^{(0)} - \lambda_1).$$

*Proof.* Since the roots of $g$ are real, we can write

$$g(x) = a_n \cdot \prod_{i=1}^{n} (x - \lambda_i)$$

with $\lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_1$. Then, we have that $g'(x) = a_n \cdot \sum_i \prod_{j \neq i} (x - \lambda_i)$ and hence

$$\frac{g(x)}{g'(x)} = \frac{1}{\sum_i \frac{1}{x - \lambda_i}}.$$

For any $x \geq \lambda_1$, we have

$$\frac{g(x)}{g'(x)} \leq \frac{1}{\frac{1}{x-\lambda_1}} = x - \lambda_1,$$
$$\frac{g(x)}{g'(x)} \geq \frac{1}{\sum_i \frac{1}{x-\lambda_1}} = \frac{x - \lambda_1}{n}.$$

Hence, by induction, we have that $x^{(k)} \geq \lambda_1$ every step and that

$$x^{(k+1)} - \lambda_1 \leq (1 - \frac{1}{n})(x^{(k)} - \lambda_1).$$

□

**Exercise 4.3.3.** Consider the following iteration:

$$x^{t+1} = x^t - \frac{1}{n^{1/k}} \frac{g^{(k-1)}(x)}{g^{(k)}(x)}$$

where $g^{(k)}(x) = \sum_i \frac{1}{(x-\lambda_i)^k}$. For $k = 1$, this is exactly the Newton iteration as $g^{(0)}(x) = n$. Show that when applied to a degree $n$ real-rooted polynomial, starting with $x^{(0)} > \lambda_1$, in each iteration the distance to the largest root decreases by a factor of $1 - \frac{1}{n^{1/k}}$.

## Quasi-Newton Method

When the Jacobian of $g$ is not available, we can approximate it using existing information. For one dimension, we can approximate the Newton method and get the following secant method

$$x^{(k+1)} = x^{(k)} - g(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{g(x^{(k)}) - g(x^{(k-1)})}$$

where we approximated the $g'(x^{(k)})$ by $\frac{g(x^{(k)}) - g(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$. For nice enough function, the convergence rate satisfies $\varepsilon_{k+1} \leq C \cdot \varepsilon_k^{\frac{1+\sqrt{5}}{2}}$, which is super linearly but not quadratic.

For higher dimension, we need to approximate the Jacobian of $g$. Let $J^{(k)}$ be the approximate Jacobian we maintained in the $k^{th}$ iteration. Similar to the secant method, we want to enforce

$$J^{(k+1)} \cdot (x^{(k)} - x^{(k-1)}) = g(x^{(k)}) - g(x^{(k-1)}).$$

In dimension higher than one, this does not uniquely define the $J^{(k+1)}$. One natural choice is to find $J^{(k+1)}$ that is closest to $J^{(k)}$ while satisfying the equation above. Solving the problem

$$\min_{J^{(k+1)} \cdot (x^{(k)} - x^{(k-1)}) = g(x^{(k)}) - g(x^{(k-1)})} \|J^{(k+1)} - J^{(k)}\|_F, \tag{4.3.1}$$

we have the update rule

$$J^{(k+1)} = J^{(k)} + \frac{y^{(k)} - J^{(k)} s^{(k)}}{\|s^{(k)}\|^2} s^{(k)\top} \tag{4.3.2}$$

where $s^{(k)} = x^{(k)} - x^{(k-1)}$ and $y^{(k)} = g(x^{(k)}) - g(x^{(k-1)})$.

**Exercise 4.3.4.** Prove that the equation (4.3.2) is indeed the minimizer of (4.3.1).

When $g$ is given by the gradient of a convex function $f$, we know that the Jacobian of $g$ satisfies $Dg = \nabla^2 f(x) \succeq 0$. Therefore, we should find impose some conditions such that $J^{(k)} \succeq 0$ for all $k$.

### BFGS algorithm

Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is one of the most popular quasi-Newton method. The algorithm maintains an approximate Hessian $J^{(k)}$ such that

- $J^{(k+1)}(x^{(k)} - x^{(k-1)}) = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$
- $J^{(k+1)}$ is close to $J^{(k)}$.
- $J^{(k)} \succ 0$.

To achieve all of these conditions, the natural optimization is

$$J^{(k+1)} \stackrel{\text{def}}{=} \arg \min_{Js=y, J=J^\top} \left\| W^{\frac{1}{2}} \left( J^{-1} - \left(J^{(k)}\right)^{-1} \right) W^{\frac{1}{2}} \right\|_F^2$$

where $s^{(k)} = x^{(k)} - x^{(k-1)}$, $y^{(k)} = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$ and $W = \int_0^1 \nabla^2 f(x^{(k-1)} + s(x^{(k)} - x^{(k-1)}))ds$ (or any $W$ such that $Wy = s$). In some sense, the $W^{-\frac{1}{2}}$ is just a correct change of variables so that the algorithm is affine invariant. Solving the equation above [20], one obtain the update

$$\left(J^{(k+1)}\right)^{-1} = (I - \rho_k \cdot s^{(k)} y^{(k)\top}) \left(J^{(k)}\right)^{-1} (I - \rho_k \cdot y^{(k)} s^{(k)\top}) + \rho_k \cdot s^{(k)} s^{(k)\top} \tag{4.3.3}$$

where $\rho_k = \frac{1}{y^{(k)\top} s^{(k)}}$. Alternatively, one can also show that [21]

$$J^{(k+1)} = \arg \min_{Js=y, J=J^\top} D_{KL}(\mathcal{N}(0, J) \| \mathcal{N}(0, J^{(k)})).$$

To implement the BFGS algorithm, it suffices to compute $\left(J^{(k)}\right)^{-1} \nabla f(x^{(k)})$. Therefore, we can directly use the recursive formula (4.3.3) to compute $\left(J^{(k)}\right)^{-1} \nabla f(x^{(k)})$ instead of maintaining $J^{(k)}$ or $\left(J^{(k)}\right)^{-1}$ explicitly.

In practice, the recursive formula $J^{(k)}$ becomes too expensive and hence one can stop the recursive formula after constant steps, which gives the limited-memory BFGS algorithm.

## ■ 4.4 Interior Point Method for Linear Programs (in progress)

In this section, we study interior point methods. In practice, this method reduces the problem of optimizing a convex function to solving less than 30 linear systems. In theory, it reduces the problem to $\tilde{O}(\sqrt{n})$ linear systems.

We start by describing interior point method for linear programs. We establish a polynomial bound and then later discuss implementation details.

### Basic Properties

We first consider the primal problem

$$(\text{P}): \qquad \min_x c^\top x \text{ subject to } Ax = b, x \geq 0$$

where $A \in \mathbb{R}^{m \times n}$. The difficulty of linear programs is the constraint $x \geq 0$. Without this constraint, we can simply solve it by a linear system. One natural idea to solve linear programs is to replace the hard constraint $x \geq 0$ by some other function. So, let us consider the regularized version of the linear program

$$(\text{P}_t): \qquad \min_x c^\top x - t \sum_{i=1}^n \ln x_i \text{ subject to } Ax = b.$$

We will explain the reason of choosing $\ln x$ in more detail later. For now, we can think it as a nice function that blows up at $x = 0$.

One can think that $-\ln x$ gives a force from every constraint $x \geq 0$ to make sure $x \geq 0$ is true. Since the gradient of $-\ln x$ blows up when $x = 0$, when $x$ is close enough, the force is large enough to counter the cost $c$. When $t \to 0$, then the problem $(\mathrm{P}_t)$ is closer to the original problem (P) and hence the minimizer of $(\mathrm{P}_t)$ is closer to a minimizer of (P).

First, we give a formula for the minimizer of $(\mathrm{P}_t)$.

**Lemma 4.4.1** (Existence and Uniqueness of central path)**.** *If the polytope $\{Ax = b, x \geq 0\}$ has an interior, then the optimum of $(\mathrm{P}_t)$ is uniquely given by*

$$\begin{aligned} xs &= t, \\ Ax &= b, \\ A^\top y + s &= c, \\ (x, s) &\geq 0 \end{aligned}$$

*where $xs = t$ is a shorthand of $x_i s_i = t$ for all $i$.*

*Proof.* The optimality condition, using dual variables $y$ for the Lagrangian of $Ax = b$ is given by

$$c - \frac{t}{x} = A^\top y.$$

Write $s_i = \frac{t}{x_i}$, to get the formula. The solution is unique because the function $-\ln x$ is strictly convex. □

**Definition 4.4.2.** We define the central path $\mathcal{C}_t = (x^{(t)}, y^{(t)}, s^{(t)})$ as the sequence of points satisfying

$$\begin{aligned} x^{(t)} s^{(t)} &= t, \\ Ax^{(t)} &= b, \\ A^\top y^{(t)} + s^{(t)} &= c, \\ (x^{(t)}, s^{(t)}) &\geq 0. \end{aligned}$$

To give another interpretation of the central path, note that the dual problem is

$$(\mathrm{D}): \qquad \max_{y,s} b^\top y \text{ subject to } A^\top y + s = c, s \geq 0.$$

Note that for any feasible $x$ and $y$, we have that

$$0 \leq c^\top x - b^\top y = c^\top x - x^\top A^\top y = x^\top s.$$

Hence, $(x, y, s)$ solves the linear program if it satisfies the central path equation with $t = 0$. Therefore, central path is a balanced way to decrease $x_i s_i$ uniformly to 0. We can formalize the intuition that for small $t$, $x^{(t)}$ is a good approximation of the primal solution.

**Lemma 4.4.3** (Duality Gap)**.** *We have that*

$$\textit{Duality Gap} = c^\top x^{(t)} - b^\top y^{(t)} = c^\top x^{(t)} - \left(x^{(t)}\right)^\top A^\top y^{(t)} = \left(x^{(t)}\right)^\top s^{(t)} = tn.$$

The interior point method follows the following framework:

1. Find $\mathcal{C}_1$
2. Until $t < \frac{\varepsilon}{n}$,
   (a) Use $\mathcal{C}_t$ to find $\mathcal{C}_{(1-h)t}$ for $h = \frac{1}{10\sqrt{n}}$.

Note that this algorithm only finds a solution with $\varepsilon$ error. If the linear program is integral, we can simply stop at small enough $\varepsilon$ and round it off to the closest integral point.

## Finding the initial point

The first question to find $\mathcal{C}_1$. This can be handled by extending the problem into higher dimension.

**Lemma 4.4.4.** *Consider a linear program $\min_{Ax=b,x\geq 0} c^\top x$ with $n$ variables and $d$ constraints. Assume that*

1. *Diameter of the polytope: For any $x \geq 0$ with $Ax = b$, we have that $\|x\|_\infty \leq R$.*
2. *Lipschitz constant of the linear program: $\|c\|_\infty \leq L$.*

*For any $0 < \delta \leq 1$, the modified linear program $\min_{\overline{A}\overline{x}=\overline{b},\overline{x}\geq 0} \overline{c}^\top \overline{x}$ with*

$$\overline{A} = \begin{bmatrix} A & 0 & \frac{1}{R}b - A1_n \\ 1_n^\top & 1 & 0 \end{bmatrix}, \overline{b} = \begin{bmatrix} \frac{1}{R}b \\ n+1 \end{bmatrix}, \text{ and } \overline{c} = \begin{bmatrix} \delta/L \cdot c \\ 0 \\ 1 \end{bmatrix}$$

*satisfies the following:*

1. $\overline{x} = \begin{bmatrix} 1_n \\ 1 \\ 1 \end{bmatrix}, \overline{y} = \begin{bmatrix} 0_d \\ -1 \end{bmatrix}$ *and* $\overline{s} = \begin{bmatrix} 1_n + \frac{\delta}{L} \cdot c \\ 1 \\ 1 \end{bmatrix}$ *are feasible primal dual vectors.*
2. *For any feasible primal dual vectors $(\overline{x}, \overline{y}, \overline{s})$ with duality gap $\leq \delta^2$, consider the vector $\hat{x} = R \cdot \overline{x}_{1:n}$ ($\overline{x}_{1:n}$ is the first $n$ coordinates of $\overline{x}$) is an approximate solution to the original linear program in the following sense*

$$\begin{aligned} c^\top \hat{x} &\leq \min_{Ax=b,x\geq 0} c^\top x + LR \cdot \delta, \\ \|A\hat{x} - b\|_1 &\leq 2n\delta \cdot \left( R \sum_{i,j} |A_{i,j}| + \|b\|_1 \right), \\ \hat{x} &\geq 0. \end{aligned}$$

*Proof.* For the first result, using $\delta < 1$, straightforward calculations show that $(\overline{x}, \overline{y}, \overline{s})$ are feasible.

For the second result, we let $\mathrm{OPT} = \min_{Ax=b,x\geq 0} c^\top x$ and $\overline{\mathrm{OPT}} = \min_{\overline{A}\overline{x}=\overline{b},\overline{x}\geq 0} \overline{c}^\top \overline{x}$. For any optimal $x$ in the original LP, $\overline{x} = \begin{bmatrix} \frac{1}{R}x \\ n+1-\frac{1}{R}\sum x_i \\ 0 \end{bmatrix}$ is feasible for the modified LP. Therefore, we have that

$$\overline{\mathrm{OPT}} \leq \frac{\delta}{L} \cdot c^\top (\frac{x}{R}) = \frac{\delta}{LR} \cdot \mathrm{OPT}.$$

Given a feasible $(\overline{x}, \overline{y}, \overline{s})$ with duality gap $\delta^2$. Write $\overline{x} = \begin{bmatrix} \overline{x}_{1:n} \\ \tau \\ \theta \end{bmatrix}$ for some $\tau \geq 0, \theta \geq 0$. We can compute $\overline{c}^\top \overline{x}$ which is $\frac{\delta}{L} \cdot c^\top \overline{x}_{1:n} + \theta$. Then, we have

$$\frac{\delta}{L} \cdot c^\top \overline{x}_{1:n} + \theta \leq \overline{\mathrm{OPT}} + \delta^2 \leq \frac{\delta}{LR} \cdot \mathrm{OPT} + \delta^2. \tag{4.4.1}$$

Hence, we can upper bound the OPT of the transformed program as follows:

$$c^\top \hat{x} = R \cdot c^\top \overline{x}_{1:n} = \frac{RL}{\delta} \cdot \frac{\delta}{L} c^\top \overline{x}_{1:n} \leq \frac{RL}{\delta} \left( \frac{\delta}{LR} \cdot \mathrm{OPT} + \delta^2 \right) = \mathrm{OPT} + LR \cdot \delta,$$

where the first step follows by $\hat{x} = R \cdot \overline{x}_{1:n}$, the third step follow by (4.4.1).

For the feasibility, we have that $\theta \leq \frac{\delta}{LR} \cdot \mathrm{OPT} + \delta^2 \leq 2n\delta$ because $\mathrm{OPT} = \min_{Ax=b,x\geq 0} c^\top x \leq nLR$. The constraint in the new polytope shows that

$$A\overline{x}_{1:n} + (\frac{1}{R}b - A1_n)\theta = \frac{1}{R}b.$$

Rewriting it, we have $A\hat{x}-b=(RA1_n-b)\theta$ and hence

$$\|A\hat{x}-b\|_1 \le \left(R\sum_{i,j}|A_{ij}|+\|b\|_1\right)\theta \le 2n\delta\cdot\left(R\sum_{i,j}|A_{ij}|+\|b\|_1\right).$$

□

## Following the central path

During the algorithm, we maintain a point $(x,y,s)$ such that $Ax=b$, $A^\top y+s=c$ and $x_is_i$ is close to $t$ for all $i$. We show how to find a feasible $(x+\delta_x,y+\delta_y,s+\delta_s)$ such that it is even closer to $t$. We can write the equation as follows:

$$\begin{aligned}(x+\delta_x)(s+\delta_s)&\approx t,\\ A(x+\delta_x)&=b,\\ A^\top(y+\delta_y)+(s+\delta_s)&=c.\end{aligned}$$

(Omitted the non-negative conditions.) Using our assumption on $(x,y,s)$ and noting that $\delta_x\cdot\delta_s$ is small, the equation can simplified as follows. We use the notation $X=\mathrm{Diag}(x)$, $S=\mathrm{Diag}(s)$.

$$\begin{bmatrix}0 & A^\top & I\\ A & 0 & 0\\ S & 0 & X\end{bmatrix}\begin{bmatrix}\delta_x\\ \delta_y\\ \delta_s\end{bmatrix}=\begin{bmatrix}0\\ 0\\ t-xs\end{bmatrix}.$$

This is a linear system and hence we can solve it exactly.

**Exercise 4.4.5.** Let $r=t-xs$. Prove that $S\delta_x=(I-P)r$ and $X\delta_s=Pr$ where $P=XA^\top(AS^{-1}XA^\top)^{-1}AS^{-1}$.

First, we show that $x^{(\mathrm{new})}=x+\delta_x$ and $s^{(\mathrm{new})}=s+\delta_s$ are feasible.

**Lemma 4.4.6.** *Suppose $\sum_i(x_is_i-t)^2\le\varepsilon^2t^2$ with $\epsilon<\frac{1}{2}$. Then,$x_i^{(\mathrm{new})}>0$ and $s_i^{(\mathrm{new})}>0$ for all $i$.*

*Proof.* Consider the orthogonal projection matrix $\overline{P}=S^{-\frac{1}{2}}X^{\frac{1}{2}}A^\top(AS^{-1}XA^\top)^{-1}AS^{-\frac{1}{2}}X^{\frac{1}{2}}$. Note that

$$X^{-1}\delta_x=S^{-\frac{1}{2}}X^{-\frac{1}{2}}(I-\overline{P})S^{-\frac{1}{2}}X^{-\frac{1}{2}}r.$$

By the assumption for each $i$, $x_is_i\ge(1-\varepsilon)t$. Therefore, we have

$$\begin{aligned}\|X^{-1}\delta_x\|_2&\le\frac{1}{\sqrt{(1-\epsilon)t}}\|(I-\overline{P})S^{-\frac{1}{2}}X^{-\frac{1}{2}}r\|_2\\ &\le\frac{1}{\sqrt{(1-\epsilon)t}}\|S^{-\frac{1}{2}}X^{-\frac{1}{2}}r\|_2\\ &\le\frac{1}{(1-\epsilon)t}\|r\|_2\le\frac{\epsilon}{1-\epsilon}.\end{aligned}$$

Similarly, we have $S^{-1}\delta_s=S^{-\frac{1}{2}}X^{-\frac{1}{2}}\overline{P}S^{-\frac{1}{2}}X^{-\frac{1}{2}}r$. Hence, we have $\|S^{-1}\delta_s\|_2\le\frac{\epsilon}{1-\epsilon}$. Therefore, when $\epsilon<\frac{1}{2}$, we have both $\|X^{-1}\delta_x\|_\infty$ and $\|S^{-1}\delta_s\|_\infty$ less than 1, which shows that both $x^{(\mathrm{new})}$ and $s^{(\mathrm{new})}$ are positive.

Next, we show that $xs$ is closer to $t$ after one step of Newton step. □

**Lemma 4.4.7.** *If $\sum_i(x_is_i-t)^2\le\varepsilon^2t^2$ with $\epsilon<\frac{1}{4}$, we have that*

$$\sum_i\left(x_i^{(\mathrm{new})}s_i^{(\mathrm{new})}-t\right)^2\le\left(\epsilon^4+16\epsilon^5\right)t^2.$$

*Proof.* We have that $x_i\delta_s+s_i\delta_x=t-x_is_i$. Using this,

$$\mathrm{LHS}=\sum_i\left(x_i^{(\mathrm{new})}s_i^{(\mathrm{new})}-t\right)^2=\sum_i(x_is_i+x_i\delta_s+s_i\delta_x+\delta_{x,i}\delta_{s,i}-t)^2=\sum_i\delta_{x,i}^2\delta_{s,i}^2\le((1+\epsilon)t)^2\cdot\sum_i\left(\frac{\delta_{x,i}}{x_i}\right)^2\left(\frac{\delta_{s,i}}{s_i}\right)^2$$

where in the last step we used $x_i^2 s_i^2 \le (1+\varepsilon)^2 t^2$. Using the previous lemma, we have that

$$\begin{aligned} \text{LHS} &\le ((1+\epsilon)t)^2 \cdot \|X^{-1}\delta_x\|_4^2 \|S^{-1}\delta_s\|_4^2 \\ &\le ((1+\epsilon)t)^2 \cdot \|X^{-1}\delta_x\|_2^2 \|S^{-1}\delta_s\|_2^2 \\ &\le ((1+\epsilon)t)^2 \left(\frac{\epsilon}{1-\epsilon}\right)^4 \\ &\le \left(\epsilon^4 + 16\epsilon^5\right) t^2 \end{aligned}$$

□

Using this, we have the main theorem.

**Theorem 4.4.8.** *We can solve a linear program to within $\delta$ "error" (see Lemma 4.4.4) in $O(\sqrt{n}\log(\frac{1}{\delta}))$ many iterations and each iteration only need to solve a linear system.*

*Proof.* Let $\Phi = \sum_i (x_i s_i - t)^2$ be the error of the current iteration. We always maintain $\Phi \le \frac{t^2}{16}$ for current $(x, y, s)$ and current $t$. Each step, we use Lemma 4.4.7 which makes $\Phi \le \frac{t^2}{50}$. Then, we decrease $t$ by $t(1 - \frac{1}{16\sqrt{n}})$. Note that

$$\sum_i (x_i s_i - t(1-h))^2 \le 2\Phi + 2t^2h^2 n \le \frac{2t^2}{50} + \frac{2t^2}{256} \le \frac{(t(1-h))^2}{16}.$$

Therefore, the invariant is preserved after each step. Since $t$ is decreased by a $(1 - \frac{1}{10\sqrt{n}})$ factor each step, it takes $O(\sqrt{n}\log(\frac{1}{\delta}))$ to decrease $t$ from 1 to $\delta^2$. □

## Why $\sqrt{n}$?

Central path is the solution to the following ODE

$$\begin{aligned} S_t \frac{d}{dt} x_t + X_t \frac{d}{dt} s_t &= 1, \\ A \frac{d}{dt} x_t &= 0, \\ A^\top \frac{d}{dt} y_t + \frac{d}{dt} s_t &= 0. \end{aligned}$$

Solving this linear system, we have that $S_t \frac{dx_t}{dt} = (I - P_t)1$ and $X_t \frac{ds_t}{dt} = P_t 1$ where $P_t = X_t A^\top (A S_t^{-1} X_t A^\top)^{-1} A S_t^{-1}$. Using that $x_t s_t = t$, we have that

$$P_t = X_t A^\top (A X_t^2 A)^{-1} A X_t = S_t^{-1} A^\top (A S_t^{-2} A)^{-1} A S_t$$

and that $X_t^{-1} \frac{dx_t}{dt} = \frac{1}{t}(I - P_t)1$ and $S_t^{-1} \frac{ds_t}{dt} = \frac{1}{t} P_t 1$. Equivalently, we have

$$\frac{d \ln x_t}{d \ln t} = (I - P_t)1 \text{ and } \frac{d \ln s_t}{d \ln t} = P_t 1.$$

Note that

$$\|P_t 1\|_\infty \le \|P_t 1\|_2 = \sqrt{n}.$$

Hence, $x_t$ and $s_t$ can change by at most a constant factor when we change $t$ by a $1 \pm \frac{1}{\sqrt{n}}$ factor.

**Exercise 4.4.9.** If we are given $x$ such that $\|\ln x - \ln x_t\|_\infty = O(1)$, then we can find $x_t$ by solving $\tilde{O}(1)$ linear systems.

## ■ 4.5 Interior Point Method for Convex Programs (in progress)

The interior point method can be used to optimize any convex function. For more in-depth treatment, please see the structural programming section in [48].

Recall that any convex optimization problem

$$\min_x f(x)$$

can be rewritten in the epigraph form as

$$\min_{\{(x,t):\ f(x)\leq t\}} t.$$

Hence, it suffices to study the problem $\min_{x\in K} c^\top x$. Similar to the case of linear programs, we replace the hard constraint $x \in K$ by a soft constraint as follows:

$$\min_x \phi_t(x) \text{ where } \phi_t(x) = tc^\top x + \phi(x).$$

where $\phi(x)$ is a convex function such that $\phi(x) \to +\infty$ as $x \to \partial K$. We call $\phi$ is a barrier for $K$. Note that we put the parameter $t$ in front of the cost $c^\top x$ instead of $\phi$ as the last lecture, it is slightly more convenient here.

To be concrete, you can always think $\phi(x) = -\sum_{i=1}^n \ln x_i$. As before, we define

**Definition 4.5.1.** The central path $x_t = \arg\min_x \phi_t(x)$.

The interior point method follows the following framework:

1. Find $x$ close to $x_1$.
2. While $t$ is not tiny,
   (a) Move $x$ closer to $x_t$
   (b) $t \to (1+h)\cdot t$.

## Self-concordant Functions

We first discuss how to move $x$ closer to $x_t$, the key part of the analysis. It would motivate the definition of $\phi$. This is simply based on Newton method:

$$x' = x - (\nabla^2\phi_t(x))^{-1}\nabla\phi_t(x).$$

A key property of Newton method is that Newton method is invariant under linear transformation. In general, whenever a method uses the $k^{th}$ order information, we need to assume the $k^{th}$ derivative is continuous in certain sense. Otherwise, the $k^{th}$ derivative is meaningless for algorithmic purpose. For Newton method, it is convenient to assume that the Hessian is Lipschitz. Since Newton method is invariant under linear transformation, it only makes sense to impose an assumption that is invariant under linear transformation.

**Exercise 4.5.2.** Compute the Newton step for the problem $\min_{x:Ax=b,} t(c^T x) - \sum_i \ln x_i$. (Note: the update has to keep the new point feasible.)

**Definition 4.5.3.** Given a convex function $f$, define $\|v\|_x^2 = v^\top \nabla^2 f(x) v$. We call a function $f$ self-concordant if for any $h \in \mathbb{R}^n$ and for any $x$ in dom$f$, we have

$$D^3 f(x)[h,h,h] \leq 2\|h\|_x^3$$

where $D^k f(x)[h_1, h_2, \cdots, h_k]$ is the directional $k^{th}$ derivative of $f$ on the direction $h_1, h_2, \cdots, h_k$.

**Exercise 4.5.4.** Equivalently, restricted on any straight line $g(t) = f(x+th)$, we have $g'''(t) \leq g''(t)^{3/2}$. Also, this is same as

$$D^3 f(x)[h_1,h_2,h_3] \leq 2\|h_1\|_x \|h_2\|_x \|h_3\|_x.$$

*Remark.* The constant 2 is chosen such that $-\ln(x)$ exactly satisfies the assumption and it is not very important.

**Example 4.5.5.** $x^\top Ax$, $-\ln x$, $-\ln(1-\sum x_i^2)$, $-\ln\det X$ are self-concordant.

By some simple calculation, one can show the following result:

**Theorem 4.5.6.** *Given a $\nu$-self concordant barrier $\phi$, we can solve $\min_{x\in K} c^\top x$ up to $\varepsilon$ error in $O^*(\sqrt{\nu})$ iterations. Each iteration involves computing $\nabla\phi(x)$ and $(\nabla^2\phi(x))^{-1}$ at certain point $x$.*

Not all convex functions are self-concordant. However, for our purpose, it suffices to show that we can construct a self-concordant barrier for any convex set.

**Theorem 4.5.7.** *Any convex set has a $O(n)$-self concordant barrier.*

Unfortunately, this is an existence result and the barrier function is expensive to compute. In practice, we construct self-concordant barriers out of simpler one:

**Lemma 4.5.8.** *We have the following self-concordant barriers. We use $\nu$-sc as a short form for $\nu$-self-concordant barrier.*

- $-\ln x$ *is* 1-*sc for* $\{x \geq 0\}$.
- $-\ln\cos(x)$ *is* 1-*sc for* $\{|x| \leq \frac{\pi}{2}\}$.
- $-\ln(t^2 - \|x\|^2)$ *is* 2-*sc for* $\{t \geq \|x\|_2\}$.
- $-\ln\det X$ *is* $n$-*sc for* $\{X \in \mathbb{R}^{n\times n}, X \succeq 0\}$.
- $-\ln x - \ln(\ln x + t)$ *is* 2-*sc for* $\{x \geq 0, t \geq -\ln x\}$.
- $-\ln t - \ln(\ln t - x)$ *is* 2-*sc for* $\{t \geq e^x\}$.
- $-\ln x - \ln(t - x\ln x)$ *is* 2-*sc for* $\{x \geq 0, t \geq x\ln x\}$.
- $-2\ln t - \ln(t^{2/p} - x^2)$ *is* 4-*sc for* $\{t \geq |x|^p\}$ *for* $p \geq 1$.
- $-\ln x - \ln(t^p - x)$ *is* 2-*sc for* $\{t^p \geq x \geq 0\}$ *for* $0 < p \leq 1$.
- $-\ln t - \ln(x - t^{-1/p})$ *is* 2-*sc for* $\{x > 0, t \geq x^{-p}\}$ *for* $p \geq 1$.
- $-\ln x - \ln(t - x^{-p})$ *is* 2-*sc for* $\{x > 0, t \geq x^{-p}\}$ *for* $p \geq 1$.

The following lemma shows how we can combine barriers:

**Lemma 4.5.9.** *If $\phi_1$ and $\phi_2$ are $\nu_1$ and $\nu_2$-self concordant barriers on $K_1$ and $K_2$ with respectively, then $\phi_1 + \phi_2$ is a $\nu_1 + \nu_2$ self concordant barrier on $K_1 \cap K_2$.*

**Lemma 4.5.10.** *If $\phi$ is a $\nu$-self concordant barrier on $K$, then $\phi(Ax)$ is a $\nu$-self concordant on $\{y : Ay \in K\}$.*

**Exercise 4.5.11.** Using the lemmas above, prove that $-\sum_{i=1}^m \ln(a_i^\top x - b_i)$ is a $m$-self concordant barrier on $\{Ax \geq b\}$.

Finally, let us consider an example:

$$\min_x \sum_{i=1}^m |a_i^\top x - b_i|^p$$

where $x \in \mathbb{R}^n$. Rewriting it, we have

$$\min_{x\in\mathbb{R}^n, s,t\in\mathbb{R}^m} \sum_i t_i \text{ subjects to } Ax - b = s, t \geq |s|^p.$$

Hence, the regularized function is simply

$$\min_{Ax-b=s} \mu\sum_i t_i - 2\sum \ln t_i - \sum_i \ln(t_i^{2/p} - s_i^2).$$

This gives an $O^*(\sqrt{m})$ iteration algorithm to solve $\ell_p$ regression.

# Chapter 5

# Geometrization and Sampling

In this chapter, we study some efficient sampling algorithms.

## Ball Walk

The simplest continuous algorithm for exploring space is Brownian motion with the ODE $dx_t = dW_t$. To turn this into a sampling algorithm, for a convex function $f$, we saw an extension using the gradient of $f$, namely

$$dx_t = -\nabla f(x_t) + \sqrt{2}dW_t$$

which can be used to sample according to the density proportional to $e^{-f}$. In this chapter we will begin with an even simpler method, which does not need access to the gradient or assume differentiability, only an oracle that can evaluate $F$.

**Algorithm 13:** `BallWalk`

```
Input: Step-size $\delta$, number of steps $T$, starting point $x_0$ in the support of target density $Q$.
Repeat $T$ times: at a point $x$,
    1. Pick a random point $y$ in the $\delta$-ball centered at $x$.
    2. Go to $y$ with probability $\min\left\{1, \frac{Q(y)}{Q(x)}\right\}$.
return $x$.
```

**Exercise 5.0.1.** Show that the distribution with density $Q$ is *stationary* for the ball walk in a connected, compact full-dimensional set, i.e., if the distribution of the current point $x$ has density $Q$, it remains $Q$.

Under mild conditions, the distribution of the current point approaches the target density $Q$. The main question is the rate of convergence, which would allow us to bound the number of steps. Note that each step involves only a function evaluation, to an oracle that outputs the value of a function proportional to the desired density. To bound the rate of convergence (and as a result the uniqueness of the stationary distribution), we first develop some general tools.

## ■ 5.1 Basics of Markov chains

For more detailed reading, including additional properties, see Section 1 of [43].

A Markov chain is defined using a $\sigma$-algebra $(K, \mathcal{A})$, where $K$ is the state space and $\mathcal{A}$ is a set of subsets of $K$ that is closed under complements and countable unions. For each element $u$ of $K$, we have a probability measure $P_u$ on $(K, \mathcal{A})$, i.e., each set $A \in \mathcal{A}$ has a probability $P_u(A)$. Informally, $P_u$ is the distribution obtained upon taking one step from $u$. The triple $(K, \mathcal{A}, \{P_u : u \in K\})$ along with a starting distribution $Q_0$ defines a Markov chain, i.e., a sequence of elements of $K$, $w_0, w_1, \ldots$, where $w_0$ is chosen from $Q_0$ and each subsequent $w_i$ is chosen from $P_{w_{i-1}}$. The choice of $w_{i+1}$ depends only on $w_i$ and is independent of $w_0, \ldots, w_{i-1}$.

A distribution $Q$ on $(K, \mathcal{A})$ is called *stationary* if taking one step from it maintains the distribution, i.e., for any $A \in \mathcal{A}$,

$$\int_K P_u(A)\, dQ(u) = Q(A).$$

A distribution $Q$ is *atom-free* if there is no $x \in K$ with $Q(x) > 0$.

**Example.** For the ball walk in a convex body, the state space $K$ is the convex body, and $A$ is the set of all measurable subsets of $K$. The next step distribution is

$$P_u(\{u\}) = 1 - \frac{\text{vol}\,(K \cap (u + \delta B_n))}{\text{vol}(\delta B_n)}$$

and for any measurable subset $A$,

$$P_u(A) = \frac{\text{vol}\,(A \cap (u + \delta B_n))}{\text{vol}(\delta B_n)} + 1_{u \in A}(u) P_u(\{u\})$$

The uniform distribution is stationary, i.e., $Q(A) = \frac{\text{vol}(A)}{\text{vol}(K)}$.

The *ergodic flow* of a subset $A$ w.r.t. the distribution $Q$ is defined as

$$\Phi(A) = \int_A P_u(K \setminus A)\, dQ(u).$$

A distribution $Q$ is stationary if and only if $\Phi(A) = \Phi(K \setminus A)$. The existence and uniqueness of the stationary distribution $Q$ for general Markov chains is a subject on its own (see e.g., [Revuz]; for Markov chains on discrete state spaces, the characterization is much simpler; see e.g., [Nor]). One way to ensure uniqueness of a stationary distribution is to use *lazy* Markov chains. In a lazy version of a given Markov chain, at each step, with probability 1/2, we do nothing; with the rest we take a step according to the Markov chain. The next theorem is folklore.

**Theorem 5.1.1.** *If $Q$ is stationary w.r.t. a lazy Markov chain then it is the unique stationary distribution for that Markov chain.*

Informally, the mixing rate of a random walk is the number of steps required to reduce some measure of the distance of the current distribution to the stationary distribution by a constant factor. The following notions will be useful for comparing two distributions $P, Q$.

1. Total variation distance is $d_{tv}(P, Q) = \sup_{A \in \mathcal{A}} |P(A) - Q(A)|$.
2. $L_2$ distance of $P$ with respect to $Q$ is

$$d_2(P, Q) = \int_K \frac{dP(u)}{dQ(u)}\, dP(u) = \int_K \left(\frac{dP(u)}{dQ(u)}\right)^2 dQ(u).$$

3. $P$ is said to be $M$-warm w.r.t. $Q$ if $M = \sup_{A \in \mathcal{A}} \frac{P(A)}{Q(A)}$.

## Convergence via Conductance

Now we introduce an important tool to bound the rate of convergence of $Q_t$,the distribution after $t$ steps to $Q$. Assume that $Q$ is the unique stationary distribution. The *conductance* of a subset $A$ is defined as

$$\phi(A) = \frac{\Phi(A)}{\min\{Q(A), Q(K \setminus A)\}}$$

and the conductance of the Markov chain is

$$\phi = \min_A \phi(A) = \min_{0 < Q(A) \le \frac{1}{2}} \frac{\int_A P_u(K \setminus A)\, dQ(u)}{Q(A)}.$$

The *local* conductance of an element $u$ is $\ell(u) = 1 - P_u(\{u\})$.

For any $0 \le s < \frac{1}{2}$, the $s$-conductance of a Markov chain is defined as

$$\phi_s = \min_{A: s < Q(A) \le \frac{1}{2}} \frac{\phi(A)}{Q(A) - s}.$$

Ideally we would like to show that $d(Q_t, Q)$, the distance between the distribution after $t$ steps and the target $Q$ is monotonically (and rapidly) decreasing. While this is not true in general for the total variation distance, it is true for a slight extension. We consider

$$\sup_{A: Q(A) = x} Q_t(A) - Q(A)$$

for each $x \in [0, 1]$. To prove inductively that this quantity decreases , Let $\mathcal{G}_x$ be the set of functions defined as

$$\mathcal{G}_x = \left\{ g : K \to [0,1] \ : \ \int_{u \in K} g(u)\ dQ(u) = x \right\}.$$

Using this, define

$$h_t(x) = \sup_{g \in \mathcal{G}_x} \int_{u \in K} g(u)\ (dQ_t(u) - dQ(u)) = \sup_{g \in \mathcal{G}_x} \int_{u \in K} g(u)\ dQ_t(u) - x.$$

This function is strongly concave.

**Exercise 5.1.2.** Show that the function $h_t$ is concave, and if $Q$ is atom-free , then $h_t(x) = \sup_{A: Q(A) = x} Q_t(A) - Q(A)$ and the supremum is achieved by some subset.

**Lemma 5.1.3.** *Let $Q$ be atom-free and $t \ge 1$. For any $0 \le x \le 1$, let $y = \min\{x, 1 - x\}$. Then,*

$$h_t(x) \le \frac{1}{2} h_{t-1}(x - 2\phi y) + \frac{1}{2} h_{t-1}(x + 2\phi y).$$

*Proof.* Assume that $0 \le x \le \frac{1}{2}$. We construct two functions, $g_1$ and $g_2$, and use these to bound $h_t(x)$. Let $A$ be a subset that achieves $h_t(x)$. Define

$$g_1(u) = \begin{cases} 2P_u(A) - 1 & \text{if } u \in A, \\ 0 & \text{if } u \notin A, \end{cases} \quad \text{and} \quad g_2(u) = \begin{cases} 1 & \text{if } u \in A, \\ 2P_u(A) & \text{if } u \notin A. \end{cases}$$

Note that $\frac{1}{2}(g_1 + g_2)(u) = P_u(A)$ for all $u \in K$, which means that

$$\frac{1}{2} \int_{u \in K} g_1(u)\ dQ_{t-1}(u) + \frac{1}{2} \int_{u \in K} g_2(u)\, dQ_{t-1}(u) = \int_{u \in K} P_u(A)\, dQ_{t-1}(u) = Q_t(A).$$

Since the walk is lazy, $P_u(A) \ge \frac{1}{2}$ iff $u \in A$, the range of the functions $g_1, g_2$ is $[0, 1]$. We let

$$x_1 = \int_{u \in K} g_1(u)\ dQ(u) \quad \text{and} \quad x_2 = \int_{u \in K} g_2(u)\ dQ(u),$$

then $g_1 \in \mathcal{G}_{x_1}$ and $g_2 \in \mathcal{G}_{x_2}$. Moreover,

$$\frac{1}{2}(x_1 + x_2) = \frac{1}{2} \int_{u \in K} g_1(u)\ dQ(u) + \frac{1}{2} \int_{u \in K} g_2(u)\ dQ(u) = \int_{u \in K} P_u(A)\ dQ(u) = Q(A) = x.$$

since $Q$ is stationary.

$$\begin{aligned} h_t(x) &= Q_t(A) - Q(A) \\ &= \frac{1}{2} \int_{u \in K} g_1(u)\ dQ_{t-1}(u) + \frac{1}{2} \int_{u \in K} g_2(u)\ dQ_{t-1}(u) - Q(A) \\ &= \frac{1}{2} \int_{u \in K} g_1(u)\ (dQ_{t-1}(u) - dQ(u)) + \frac{1}{2} \int_{u \in K} g_2(u)\ (dQ_{t-1}(u) - dQ(u)) \\ &\le \frac{1}{2} h_{t-1}(x_1) + \frac{1}{2} h_{t-1}(x_2). \end{aligned}$$

Next,

$$\begin{aligned}
x_1 &= \int_{u\in K} g_1(u)\ dQ(u) \\
&= 2\int_{u\in A} P_u(A)\ dQ(u) - \int_{u\in A} dQ(u) \\
&= 2\int_{u\in A} (1-P_u(K\setminus A))\ dQ(u) - x \\
&= x - 2\int_{u\in A} P_u(K\setminus A)\ dQ(u) \\
&= x - 2\Phi(A) \\
&\le x - 2\phi x \\
&= x(1-2\phi).
\end{aligned}$$

Thus we have, $x_1 \le x(1-2\phi) \le x \le x(1+2\phi) \le x_2$. Since $h_{t-1}$ is concave, the chord from $x_1$ to $x_2$ on $h_{t-1}$ lies below the chord from $[x(1-2\phi), x(1+2\phi)]$. Therefore,

$$h_t(x) \le \frac{1}{2}h_{t-1}(x(1-2\phi)) + \frac{1}{2}h_{t-1}(x(1+2\phi)).$$

□

A proof along the same lines implies the following generalization.

**Lemma 5.1.4.** *Let $Q$ be atom-free and $0 \le s \le 1$. For any $s \le x \le 1-s$, let $y = \min\{x-s, 1-x-s\}$. Then for any integer $t > 0$,*

$$h_t(x) \le \frac{1}{2}h_{t-1}(x - 2\phi_s y) + \frac{1}{2}h_{t-1}(x + 2\phi_s y).$$

These results can be extended to the case when $Q$ has atoms with slightly weaker bounds [43].

**Theorem 5.1.5.** *Let $0 \le s \le 1$ and $C_0$ and $C_1$ be such that*

$$h_0(x) \le C_0 + C_1 \min\{\sqrt{x-s}, \sqrt{1-x-s}\}.$$

*Then*

$$h_t(x) \le C_0 + C_1 \min\{\sqrt{x-s}, \sqrt{1-x-s}\}\left(1 - \frac{\phi_s^2}{2}\right)^t.$$

The proof is by induction on $t$.

**Corollary 5.1.6.** *We have*

1. *Let $M = \sup_A Q_0(A)/Q(A)$. Then,*

$$d_{TV}(Q_t, Q) \le \sqrt{M}\left(1 - \frac{\phi^2}{2}\right)^t.$$

2. *Let $0 < s \le \frac{1}{2}$ and $H_s = \sup\{|Q_0(A) - Q(A)| : Q(A) \le s\}$. Then,*

$$d_{TV}(Q_t, Q) \le H_s + \frac{H_s}{s}\left(1 - \frac{\phi_s^2}{2}\right)^t.$$

3. *Let $M = d_2(Q_0, Q)$. Then for any $\varepsilon > 0$,*

$$d_{TV}(Q_t, Q) \le \varepsilon + \sqrt{\frac{M}{\varepsilon}}\left(1 - \frac{\phi^2}{2}\right)^t.$$

### Convergence via Log-Sobolev

For a warm start, the convergence rate established by conductance is asymptotically optimal in many cases of interest, including the ball walk for convex body. However, when the starting distribution is more focused, e.g., a single point, then there is a significant starting penalty usually a factor of the dimension or larger. One way to avoid this is to observe that the conductance of smaller subsets is in fact even higher and that one does not need to pay this large starting penalty. A classical technique in this regard is the log-Sobolev constant. For a Markov chain with stationary density $Q$ and transition operator $P$, we can define it as follows.

$$\rho = \inf_{g\text{:smooth}, \int g(x)^2 dQ(x) = 1} \frac{\int_{x,y\in K} (f(x)-f(y))^2 P(x,y) dQ(x)}{\int f(x)^2 \log f(x)^2 dQ(x)}.$$

This parameter allows us to show convergence of the current distribution to the target in relative entropy. Recall that the relative entropy of a distribution $P$ with respect to a distribution $Q$ is

$$H_Q(P) = \int_K P(x) \log \frac{P(x)}{Q(x)} dQ(x).$$

**Theorem 5.1.7.** *For a Markov chain with distribution $Q_t$ at time $t$, and log-Sobolev parameter $\rho$, we have*

$$H_Q(Q_t) \le e^{-2\rho t} H_Q(Q_0).$$

## ■ 5.2 Conductance of the Ball Walk

In the section we bound the conductance of the ball walk when applied to the indicator function of a convex body. At first glance, the ball walk is not an efficient algorithm, even for uniformly sampling a convex body. The reason is simply that the local conductance could be exponentially small (consider a point close to the vertex of a polyhedron). We can get around this in two ways. The first, which is simpler, but less efficient is to "smoothen" the convex body by taking the Minkowski sum with a small Euclidean ball, i.e., replace $K$ with $K + \alpha B^n$.

**Exercise 5.2.1.** Let $K$ be a convex body in $\mathbb{R}^n$ containing the unit ball. Show that (a) $\text{vol}(K+\alpha B^n) \le (1+\alpha)^n K$ and (b) with $\delta = \alpha/\sqrt{n}$ the local conductance of every point in $K + \alpha B^n$ is at least an absolute constant.

Using the exercise, it suffices to set $\alpha = 1/n$, so that a sample from $K + \alpha B^n$ has a large probability of being in $K$ and then $\delta = 1/n^{3/2}$.

The second approach is to show that the local conductance is in fact large almost everywhere, and if the starting distribution is "warm" then these points can effectively be ignored. This will allow us to make $\delta$ much larger, namely $\delta = 1/\sqrt{n}$. Larger step sizes should allows us to prove faster mixing.

To convey the main ideas of the analysis, we focus on the first approach here. The goal is to show that the conductance of any subset is large, i.e., the probability of crossing over in one step is at least proportional to the measure of the set or its complement, whichever is smaller. First, we argue that the one-step distributions of two points will have a significant overlap if the points are sufficient close.

**Lemma 5.2.2** (One-step overlap)**.** *Let $u, v \in K$ s.t. $\ell(u), \ell(v) \ge \ell$ and $\|u - v\| \le \frac{t\delta}{\sqrt{n}}$. Then the one-step distributions from them satisfy $d_{TV}(P_u, P_v) \le 1 + t - \ell$.*

Setting $t = \ell/2$, this says that if the total variation distance between the one-step distributions from $u, v$ is greater than $1 - \ell/2$, then the distance between them is at least $\frac{\ell\delta}{2\sqrt{n}}$. What this effectively says is that points close to the internal boundary of a subset are likely to cross over to the other side. To complete a proof we would need to show that the internal boundary of any subset is large if the subset (or its complement) is large, a purely geometric property.

**Theorem 5.2.3** (Isoperimetry)**.** *Let $S_1, S_2, S_3$ be a partition of a convex body $K$ of diameter $D$. Then,*

$$\text{vol}(S_3) \ge \frac{2}{D} d(S_1, S_2) \min\{\text{vol}(S_1), \text{vol}(S_2)\}.$$

This can be generalized to any logconcave measure. We will discuss this and other extensions in detail later. But first we bound the conductance.

**Theorem 5.2.4.** *Let $K$ be a convex body in $\mathbb{R}^n$ of diameter $D$ containing the unit ball and with every $u \in K$ having $\ell(u) \geq \ell$. Then the conductance of the ball walk on $K$ with step size $\delta$ is*

$$\Omega\left(\frac{\ell^2\delta}{\sqrt{n}D}\right).$$

**Corollary 5.2.5.** *The ball walk in a convex body with local conductance at least $\ell$ everywhere has mixing rate $O(nD^2\delta^2/\ell^4)$.*

Using the construction above of adding a small ball to every point of $K$ gives a lower bound of $\delta = 1/n^{3/2}$ and $\ell = \Omega(1)$ and thus a polynomial bound of $O(n^4D^2)$ on the mixing time.

## Warm Start and $s$-Conductance

A more careful analysis, using a warm start and only bounding the conductance of large sets gives a better (and tight) bound for the mixing of the ball walk.

**Theorem 5.2.6.** *From a warm start, the ball walk in a convex body of diameter $D$ containing a unit ball has a mixing rate of $O(n^2D^2)$ steps.*

This is based on two ideas: (1) most points of a convex body containing a unit ball have large local conductance and we can use $\delta = 1/\sqrt{n}$ instead of $1/n^{3/2}$, (2) the $s$-conductance is large and hence the walk mixes from a suitably warm start.

**Lemma 5.2.7.** *Let $K$ be a convex body containing a unit ball. For the ball walk with $\delta$ step size, let $K_\delta = \left\{u \in K : \ell(u) \geq \frac{3}{4}\right\}$. Then $K_\delta$ is a convex set and $\mathrm{vol}(K_\delta) \geq (1 - 2\delta\sqrt{n})\mathrm{vol}(K)$.*

**Exercise 5.2.8.** Prove the first part of the previous lemma.

**Theorem 5.2.9.** *The $s$-conductance of the ball walk with $\delta = \frac{s}{4\sqrt{n}}$ step size in a convex body $K$ of diameter $D$ and containing the unit ball satisfies*

$$\phi_s \gtrsim \frac{s}{nD}.$$

The mixing rate follows by applying Theorem 5.1.5 and Corollary 5.1.6.

## Tightness of the bound

The mixing rate of $O(n^2D^2)$ for the ball walk is in fact the best possible even from a warm start (with the assumption of a unit ball inside the convex body and diameter $D$). To see this, consider a cylinder whose cross-section is a unit ball and axis is $[0, D]$ along $e_1$. Suppose the starting distribution is uniform in the part of the cylinder in $[0, D/3]$. Then we claim that to reach the opposite third of the cylinder needs $\Omega(n^2D^2)$ steps with high probability. Each step has length at most $\delta$ in a random direction, and this is about $\delta/\sqrt{n}$ along $e_1$. Viewing this as an unbiased random walk along $e_1$, the effective diameter is $D/(\delta/\sqrt{n})$ and hence the number of steps to cross an interval of length $D/3$ is $\Omega(nD^2/\delta^2) = \Omega(n^2D^2)$.

**Exercise 5.2.10.** Prove the above claim rigorously.

## Speedy walk

In the above analysis of the ball walk, the dependence on the error parameter $\varepsilon$, the distance to the target distribution, is polynomial in $1/\varepsilon$ rather than its logarithm. The speedy walk is a way to improve the analysis. In the speedy walk, at a point $x$, we sample uniformly from the intersection of $(x + \delta B^n) \cap K$ and go there. The resulting Markov chain is the subsequence of *proper* steps of the ball walk.

**Theorem 5.2.11.** *The conductance of the speedy walk is $\Omega(1/nD)$.*

To analyze the ball walk, we then need to show that the number of "wasted" steps is not too many. This follows from the assumption of a wam start and Lemma 5.2.7.

## 5.3 Generating a warm start

To get the mixing rate of $O(n^2D^2)$, we need a warm start, i.e., a distribution whose density at any point is within $O(1)$ of the target density.

**Algorithm 14:** `Warm Start`

```
Input: membership oracle for $K$ s.t. $B^n \subseteq K \subseteq DB^n$.
Let $x$ be a random point in $B^n$. Define $K_i = 2^{i/n}B^n \cap K$.
for $i = 1, \cdots, n\log D$ do
    1. Use ball walk from $x$ to generate random point $y$ in $K_i$.
    2. Set $x = y$.
end
return $x$.
```

Since $K_{i+1} \subseteq 2^{1/n}K_i$, we have $\text{vol}(K_{i+1}) \leq 2\text{vol}(K_i)$ and hence a 2-warm start is maintained. Once we have a random point from $K$, subsequent random points can be generated by simply continuing the ball walk; thus the cost of the warm start is only for the first sample.

## 5.4 Isotropic Transformation

The complexity of sampling with the ball walk is polynomial in $n, D$ and $\log(1/\varepsilon)$ to get within $\varepsilon$ of the target density. This is not a polynomial algorithm since the dependence is on $D$ and not $\log D$. To get a polynomial algorithm, we need one more ingredient.

We say that a distribution $Q$ is *isotropic* if $\mathbf{E}_Q(x) = 0$ and $\mathbf{E}_Q(xx^T) = I$, i.e., the mean is zero and the covariance matrix (exists and) is the identity. We say that the distribution is $C$-isotropic if the eigenvalues of its covariance matrix are in $[\frac{1}{C}, C]$. An affine transformation is said to be an *isotropic tranformation* if the resulting distribution is isotropic.

Any distribution with bounded second moments has an isotropic transformation. It is clear that satisfying the first condition is merely a translation, so assume the mean is zero. For the second, suppose the covariance matrix is $\mathbf{E}_Q(xx^T) = A$. Then consider $y = A^{-1/2}x$. It is easy to see that

$$\mathbf{E}(yy^T) = A^{-1/2}\mathbf{E}\left(xx^T\right)A^{-1/2} = I.$$

For convex bodies, isotropic position comes with a strong guarantee.

**Theorem 5.4.1.** *For a convex body in isotropic position (i.e., the uniform distribution over the body is isotropic), we have*

$$\sqrt{\frac{n+2}{n}}B^n \subseteq K \subseteq \sqrt{n(n+2)}B^n.$$

Thus the effective diameter is $O(n)$. If we could place a convex body in isotropic position before sampling, we would have a poly$(n)$ algorithm. In fact, it is even better than this as most points are within distance $O(\sqrt{n})$ of the center of gravity. We quote a theorem due to Paouris.

**Theorem 5.4.2.** *For an isotropic logconcave density $p$ in $\mathbb{R}^n$ and any $t \geq 1$,*

$$\Pr_p\left(\|x\| \geq ct\sqrt{n}\right) \leq e^{-t\sqrt{n}}.$$

How to compute an isotropic transformation? This is easy, from the definition, all we need is to estimate its covariance matrix, which can be done from random samples. Thus, if we could sample $K$, we can compute an isotropic transformation for it. This appears cyclic – we need isotropy for efficient sampling and efficient sampling for isotropy. The solution is simply to bootstrap them.

**Algorithm 15:** `IsotropicTransform`

```
Input: membership oracle for $K$ s.t. $B^n \subseteq K \subseteq DB^n$.
Let $x$ be a random point in $B^n$, $A = I$ and $K_i = 2^{i/n}B^n \cap K$.
for $i = 1, \cdots, n \log D$ do
    1. Use the ball walk from $x$ to generate $N$ random points $x_1 \ldots x_N$ in $AK_i$.
    2. Compute $C = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$ and set $A = C^{-1/2}A$.
    3. Set $x = x_N$.
end
return $x$.
```

We will choose $N$ large enough so that after the transformation $K_i$ is 2-isotropic and therefore $K_{i+1}$ is 6-isotropic. We can bound $N$ as follows.

**Exercise 5.4.3.** Show that if $K$ is isotropic, then with $N = O(n^2)$, the matrix $A = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$ for $N$ random samples from $K$ satisfies $\|A - I\|_{\text{op}} \le 0.5$.

A tight bound on the sample complexity was established by [1].

**Theorem 5.4.4.** *For an isotropic logconcave distribution $Q$ in $\mathbb{R}^n$, the covariance $N = O(n)$ random samples satisfies $\|A - I\|_{\text{op}} \le 0.5$.*

Thus the overall algorithm needs $O(n \log D)$ phases, with $O(n)$ samples in each phase from a near-isotropic distribution, and thus poly$(n)$ steps per sample.

## 5.5 The role of isoperimetry

Theorem 5.2.3 was refined by KLS [26] as follows (we state it here for logconcave densities).

**Theorem 5.5.1.** *For any partition $S_1, S_2, S_3$ of $\mathbb{R}^n$, and any logconcave measure $\mu$ in $\mathbb{R}^n$,*

$$\mu(S_3) \ge \frac{\ln 2}{\mathbf{E}_\mu(\|x - \bar{x}\|)} \min\{\mu(S_1), \mu(S_2)\}.$$

Thus, for a (near-)isotropic distribution, the diameter can be replaced by $O(\sqrt{n})$ and this gives a bound of $O(n^3)$ from a warm start. One way to summarize the analysis so far is that the complexity of sampling a convex body (and in fact a logconcave density) from a warm start is $O^*(n^2/\psi^2)$ where $\psi$ is the isoperimetric ratio of the convex body. In other words, the expansion of the Markov chain reduces to the expansion of the target logconcave density. It then becomes a natural question to find the best possible estimate for the isoperimetric ratio. KLS also provided a conjecture for this.

**Conjecture 5.5.2.** *The isoperimetric ratio of any isotropic logconcave density in $\mathbb{R}^n$ is $\Omega(1)$.*

The bound of the conjecture holds for all halfspace induced subsets. So the conjecture says that the worst isoperimetry is achieved up to a constant factor by a halfspace (this version does not need isotropic position).

## 5.6 Hit-and-Run

The ball walk does not mix rapidly from some starting points. While this hurdle can be overcome by starting with a deep point and carefully maintaining a warm start, it is natural to ask if there is a simple process that does mix rapidly from any starting point. Hit-and-Run satisfies this requirement.

**Algorithm 16:** `Hit − and − Run`

```
Input: starting point $x_0$ in a convex body $K$.
Repeat $T$ times: at current point $x$,
    1. Pick a uniform random direction $\ell$ through $x$.
    2. Go to uniform random point $y$ on the chord of $K$ induced by $\ell$.
return $x$.
```

Since hit-and-run is a symmetric Markov chain, the uniform distribution on $K$ is stationary for it. To sample from a general density proportional to $f(x)$, in Step 2, we sample $y$ according to the density proportional to $f$ restricted to the random line $\ell$.

Next we give a formula for the next step distribution from a point $u$.

**Lemma 5.6.1.** *The next step distribution of Hit-and-Run from a point $u$ is given by*

$$P_u(A) = \frac{2}{\mathrm{vol}(S^{n-1})} \int_A \frac{dx}{\|x-u\|^{n-1}\,\ell(u,x)}$$

*where $A$ is any measurable subset of $K$ and $\ell(u,x)$ is the length of the chord in $K$ through $u$ and $x$.*

**Exercise 5.6.2.** Prove Lemma 5.6.1.

The main theorem of this section is the following [41].

**Theorem 5.6.3.** *[41]The conductance of Hit-and-Run in a convex body $K$ containing the unit ball and of diameter $D$ is $\Omega\left(\frac{1}{nD}\right)$.*

This implies a mixing time of $O\left(n^2D^2\log(M/\varepsilon)\right)$ to get to within distance $\varepsilon$ of the target density starting from an $M$-warm initial density. By taking one step from the initial point, we can bound $M$ by $(D/d)^n$ where $d$ is the minimum distance of the starting point from the boundary. Hence this gives a bound of $\tilde{O}(n^3D^2)$ from *any* interior starting point.

The proof of the theorem follows the same high-level outline as that of the ball walk, needing two major ingredients.

Define the "median" step-size function $F$ as the $F(x)$ such that

$$\Pr(\|x-y\| \le F(x)) = \frac{1}{8}$$

where $y$ is a random step from $x$.

We also need a non-Euclidean notion of distance, namely the classical cross-ratio distance. For points $u, v$ in $K$, inducing a chord $[p,q]$ with these points in the order $p, u, v, q$, the cross-ratio distance is

$$d_K(u,v) = \frac{\|u-v\|\,\|p-q\|}{\|p-u\|\,\|v-q\|}.$$

It is related to the Hilbert distance (which is a true distance) as follows:

$$d_H(u,v) = \ln\left(1 + d_K(u,v)\right).$$

The first ingredient shows that if two points are close geometrically, then their next-step distributions have significant overlap.

**Lemma 5.6.4.** *For two points $u, v \in K$ with*

$$d_K(u,v) < \frac{1}{8} \text{ and } \|u-v\| \le \frac{2}{\sqrt{n}}\min\{F(u), F(v)\}$$

*we have $d_{TV}(P_u, P_v) < 1 - \frac{1}{500}$.*

The second ingredient is an isoperimetric inequality (independent of any algorithm). The cross-ratio distance has a nice isoperimetry inequality.

**Theorem 5.6.5.** *For any partition $S_1, S_2, S_3$ of a convex body $K$,*

$$\mathrm{vol}(S_3) \geq d_K(S_1, S_2)\frac{\mathrm{vol}(S_1)\mathrm{vol}(S_2)}{\mathrm{vol}(K)}.$$

However, this will not suffice to prove a bound on the conductance of all subsets. The reason is that we cannot guarantee a good lower bound on the *minimum* distance between subsets $S_1, S_2$. Instead, we will need a weighted isoperimetric inequality, which uses an average distance.

**Theorem 5.6.6.** *Let $S_1, S_2, S_3$be a partition of a convex body $K$. Let $h : K \to \mathbb{R}_+$ be a function s.t. for any $u \in S_1$,$v \in S_2$, and any $x$ on the chord through $u$ and $v$, we have*

$$h(x) \leq \frac{1}{3}\min\{1, d_K(u, v)\}.$$

*Then,*

$$\mathrm{vol}(S_3) \geq \mathbf{E}_K(h(x))\min\{\mathrm{vol}(S_1), \mathrm{vol}(S_2)\}.$$

For bounding the conductance, we will use a specific function $h$. To introduce it, we first define a step-size function $s(x)$:

$$s(x) = \sup\left\{t : \frac{\mathrm{vol}(x + tB^n \cap K}{\mathrm{vol}(tB^n)} \geq \gamma\right\}$$

for some fixed $\gamma \in (0, 1]$.

**Exercise 5.6.7.** Show that the step-size function is concave over any convex body.

In the proof of Theorem 5.6.3, we will set $h(x) = \frac{s(x)}{48\sqrt{n}D}$.

## 5.7 Dikin walk

Both the ball walk and hit-and-run have a dependence on the "roundness" of the target distribution, e.g., via its diameter or average distance to the center of gravity. Reducing this dependence to logarithmic by rounding is polynomial time but expensive. The current best rounding algorithm (which achieves near-isotropic position) has complexity $O^*(n^4)$ [44]. Here we describe a different approach, which is affine-invariant, but requires more knowledge of the convex body. In particular, we will focus on the special case of sampling an explicit polytope $P = \{x : Ax \geq b\}$.

For a point $x$ in the interior of $P$, let

$$\phi(x) = -\sum_{i=1}^{m}\ln(A_i x - b_i)$$

be the logarithmic barrier function that we already encontered in the Interior Point Method for linear optimization. Recall the following.

**Fact 5.7.1.** *For the log barrier $\phi$, we have*

$$\nabla\phi(x) = -A^T s(x)^{-1} \text{ and } \nabla^2\phi(x) = A^T S(x)^{-2} A$$

*where $s(x) = Ax - b$ and $S(x) = \mathrm{Diag}(s(x))$.*

We are now ready to define the Dikin walk for the log barrier. Let $\Sigma_x = \frac{1}{100n}(A^T S(x)^{-2}A)^{-1}$ and $\gamma_x(y)$ be the density of $y$ for the Gaussian $N(x, \Sigma_x)$.

```
Algorithm 17: DikinWalk
Input: starting point $x_0$ in a polytope $P = \{x : Ax \geq b\}$.
Repeat $T$ times: at current point $x$,
    1. Pick $y$ from $N(x, \Sigma_x)$.
    2. If $y \in P$ go to $y$ with probability $\min\left\{1, \frac{\gamma_y(x)}{\gamma_x(y)}\right\}$.
return $x$.
```

Each step of the walk can be implemented in matrix multiplication time $O(mn^{w-1})$. Due to the Metropolis filter, we have that the uniform distribution in $P$ is stationary for this walk. The main theorem of this section is a bound on its rate of convergence, originally proved in [27].

**Theorem 5.7.2.** *For a polytope with $m$ facets in $\mathbb{R}^n$, the conductance of the Dikin walk is $\Omega(1/\sqrt{mn})$.*

Thus the mixing time from a warm start is $O(mn)$ steps and a factor of $\tilde{O}(n)$ higher from any start. The notable feature of this guarantee is that it is affine-invariant and does not depend on the coefficients of $A, b$. Thus, for the Dikin walk, unlike the ball walk and hit-and-run, there is no need for rounding to obtain a polytime sampling algorithm. As in previous analysis, the proof is based on two crucial lemmas: a geometric isoperimetric inequality and an analysis of the overlap of next-step distributions from "nearby" points. We have already seen the former, it is Theorem 5.6.5. To apply it, we need a simple property of the Dikin walk.

For a matrix $A$, let $\|x\|_A^2 = x^T Ax$. We recall that the Dikin ellipsoid at $u \in K$ is given by $E_u = \left\{x : (x-u)^T \nabla^2\phi(u)(x-u) \leq 1\right\}$. For convenience we write $\|.\|_{\nabla^2\phi(x)}$ as $\|.\|_x$. The Dikin ellipsoid has the following property.

**Lemma 5.7.3.** *For any $u \in P$, defined by $m$ inequalities and the log barrier function $\phi : P \to \mathbb{R}$,*

$$u + E_u \subseteq P \cap (u - P) \subseteq u + \sqrt{m}E_u.$$

*Proof.* By the definition of $E_u$, for any $v \in P \cap (u-P)$, $\left\|A^T S(u)^{-1}(u-v)\right\|_\infty \leq 1$. Hence, $\left\|(u-v)A^T S(u)^{-2}A(u-v)\right\| \leq m$, as claimed. □

We will use this to prove a suitable isoperimetry.

**Lemma 5.7.4.** *For $u, v \in P$,*

$$d_P(u,v) \geq \frac{\|u-v\|_u}{\sqrt{m}}.$$

*Proof.* Consider the Dikin ellipsoid at $u$. For the chord $[p,q]$ induced by $u, v$ with these points in the order $p, u, v, q$, suppose that $\|p-u\|_2 \leq \|v-q\|_2$. Then by Lemma 5.7.3, $p \in K \cap (u-K)$. And hence $\|p-u\|_u \leq \sqrt{m}$. Therefore,

$$d_P(u,v) = \frac{\|u-v\|_2\|p-q\|_2}{\|p-u\|_2\|v-q\|_2} \geq \frac{\|u-v\|_2}{\|p-u\|_2} = \frac{\|u-v\|_u}{\|p-u\|_u} \geq \frac{\|u-v\|_u}{\sqrt{m}}.$$

□

The following one-step coupling lemma uses the same norm.

**Exercise 5.7.5.** Show that $\nabla \ln\det(A^T S(x)^{-2}A) = -2A^T S(x)^{-1}\sigma$, where $\sigma$ is the leverage score vector of $S(x)^{-1}A$ with components $\sigma_i = (S(x)^{-1}A)_i(A^T S(x)^{-2}A)^{-1}(S(x)^{-1}A)_i^T$.

**Lemma 5.7.6.** *For two points $x, y \in P$, with $\|x-y\|_x \leq \frac{1}{10\sqrt{n}}$, we have $d_{TV}(P_x, P_y) \leq \frac{3}{4}$.*

*Proof.* We have to prove two things: first, the one-step overlap of the Gaussians $\gamma_x, \gamma_y$ used by the Dikin walk at $u, v$ have large overlap and second, the rejection probability is small. Note that

$$\gamma_x(y) = \frac{1}{\sqrt{\det(\Sigma_x)}} e^{-(y-x)^T \Sigma_x^{-1}(y-x)/2}$$

where $\Sigma_x = \frac{1}{100n}(A^T S(x)^{-2} A)^{-1}$. For the first we can use Pinsker's inequality

$$d_{TV}(\gamma_x, \gamma_y) \le \sqrt{\frac{1}{2} d_{KL}(\gamma_x | \gamma_y)}.$$

To bound the KL-divergence, we write (see exercise below)

$$2d_{KL}(N(x,\Sigma_x)|N(y,\Sigma_y)) = \mathrm{tr}(\Sigma_y^{-1}\Sigma_x) - n - \ln\frac{\det(\Sigma_x)}{\det(\Sigma_y)} + (x-y)^T \Sigma_y^{-1}(x-y).$$

We start with the first term:

$$\begin{aligned}
\mathrm{tr}(\Sigma_y^{-1}\Sigma_x) &= \mathrm{tr}(A^T S(y)^{-2} A (A^T S(x)^{-2} A)^{-1}) \\
&= \mathrm{tr}(S(y)^{-2} A (A^T S(x)^{-2} A)^{-1} A^T S(y)^{-1}) \\
&= \sum_{i=1}^m (S(x)S(y)^{-1}(S(x)^{-1} A (A^T S(x)^{-2} A)^{-1} A^T S(x)^{-1}) S(x) S(y)^{-1})_{ii} \\
&= \sum_{i=1}^m \sigma_i \left(\frac{s_i(x)}{s_i(y)}\right)^2.
\end{aligned}$$

where

$$\sigma_i = (S(x)^{-1} A (A^T S(x)^{-2} A)^{-1} A^T S(x)^{-1})_{ii}$$

is the leverage score of the $i$'th row of $S(x)^{-1}A$.

Next, consider

$$f(x) = \ln\det(A^T S(x)^{-2} A).$$

We have (see exercise above)

$$\nabla f(x) = -2A^T S(x)^{-1} \sigma.$$

Since $f$ is a convex function,

$$\begin{aligned}
\ln\frac{\det(\Sigma_x)}{\det(\Sigma_y)} &= f(y) - f(x). \\
&\ge \langle \nabla f(x), y - x\rangle \\
&= -2\sum_{i=1}^m \sigma_i \frac{A(y-x)}{s(x)_i} \\
&= 2\sum_{i=1}^m \sigma_i \frac{s(x)_i - s(y)_i}{s(x)_i}
\end{aligned}$$

Therefore, using the fact that $\sigma_i \in [0,1]$, and defining

$$\Delta_i = (s(x)_i - s(y)_i)/s(x)_i,$$

we have,.

$$\begin{aligned}
\operatorname{tr}(\Sigma_y^{-1}\Sigma_x) - n - \ln\frac{\det(\Sigma_x)}{\det(\Sigma_y)} &\le \sum_{i=1}^{n}\sigma_i\left(\frac{s(x)_i}{s(y)_i}\right)^2 - n - 2\sum_{i=1}^{m}\sigma_i\frac{s(x)_i - s(y)_i}{s(x)_i}\\
&\le \sum_{i=1}^{m}\sigma_i\left(\left(\frac{s(x)_i}{s(y)_i}\right)^2 - 1 - 2\left(\frac{s(x)_i - s(y)_i}{s(x)_i}\right)\right)\\
&\le \sum_{i=1}^{m}\left(\frac{1}{(1-\Delta_i)^2} - 1 - 2\Delta_i\right)\\
&\le \sum_{i=1}^{m}\frac{3\Delta_i^2}{(1-\Delta_i)^2}\\
&\le 3\frac{\sum_{i=1}^{m}\Delta_i^2}{(1-\|\Delta\|_\infty)^2}\\
&\le 3\frac{\|x-y\|_x^2}{(1-\|x-y\|_x)^2}
\end{aligned}$$

In the last step, we used the observation that

$$\|x-y\|_x^2 = \sum_{i=1}^{m}\Delta_i^2.$$

Thus,

$$d_{TV}(\gamma_x,\gamma_y) \le \sqrt{\frac{1}{2}d_{KL}(\gamma_x:\gamma_y)} \le \sqrt{\frac{4}{2\cdot 2}}\cdot\frac{\|x-y\|_x}{1-\|x-y\|_x} \le \frac{0.2}{\sqrt{n}}.$$

Next we bound the rejection probability. This part will impose the stricter condition; the above only needs $\|x-y\|_x$ is bounded by a constant, not $O(1/\sqrt{n})$. Using the expression for $\gamma_x(y)$ and the definition of $\Sigma_x$,

$$\begin{aligned}
2\ln\left(\frac{\gamma_y(x)}{\gamma_x(y)}\right) &= \ln\frac{\det(A^TS_y^{-2}A)}{\det(A^TS_x^{-2}A)} - 100n(y-x)^TA^TS(y)^{-2}A(y-x) + 100n(y-x)^TA^TS(x)^{-2}A(y-x)\\
&\ge \langle -2A^TS(x)^{-1}\sigma, y-x\rangle - 100n(y-x)^TA^TS(y)^{-2}A(y-x) + 100n(y-x)^TA^TS(x)^{-2}A(y-x).\\
&\ge -2\sqrt{(y-x)^TA^TS(x)^{-1}\mathrm{Diag}(\sigma^2)S(x)^{-1}A(y-x)} - 100n\left\|y-x\right\|_y^2 + 100n\left\|y-x\right\|_x^2\\
&\ge \frac{-2}{10\sqrt{n}}\sqrt{2\sum_{i=1}^{m}\sigma_i^2} - \frac{0.2}{n}\\
&\ge -0.4.
\end{aligned}$$

We used two facts above. First, $\left\|S(x)^{-1}A(y-x)\right\|_\infty \le \left\|S(x)^{-1}A(y-x)\right\| = \|y-x\|_x$. Next, since $y$ is a random step from $x$, $\mathbf{E}\,\|y-x\|_x^2 \le \frac{1}{100n}$ and hence with very high probability, by Gaussian concentration,$\|y-x\|_x^2 \le \frac{2}{100n}$. Using this,

$$(y-x)^TA^TS(x)^{-1}\mathrm{Diag}(\sigma^2)S(x)^{-1}A(y-x) \le \frac{\sum_{i=1}^{m}\sigma_i^2}{100n} \le \frac{1}{100}.$$

Second, $\|y-x\|_y \le \frac{\|y-x\|_x}{(1-\|y-x\|_x)} \le \|y-x\|_x\left(1+\frac{0.2}{\sqrt{n}}\right)$

Thus, the acceptance probability is at least $e^{-0.2}$ and we conclude that $d_{TV}(P_u,P_v) < \frac{3}{4}$. □

**Exercise 5.7.7.** Show that

$$2d_{KL}(N(x,\Sigma_x)|N(y,\Sigma_y)) = \operatorname{tr}(\Sigma_y^{-1}\Sigma_x) - n - \ln\frac{\det(\Sigma_x)}{\det(\Sigma_y)} + (x-y)^T\Sigma_y^{-1}(x-y).$$

We can now prove the main conductance bound.

*Proof of Theorem 5.7.2.* We follow the standard high-level outline. Consider any measurable subset $S_1 \subseteq P$ and let $S_2 = P \setminus S_1$ be its complement. Define the points with low escape probability for these subsets as

$$S_i' = \left\{ x \in S_i : P_x(P \setminus S_i) < \frac{1}{8} \right\}$$

and $S_3' = P \setminus S_1' \setminus S_2'$. Then, for any $u \in S_1'$, $v \in S_2'$, we have $d_{TV}(P_u, P_v) > 1 - \frac{1}{4}$. Hence, by Lemma 5.7.6, we have

$$\|u - v\|_{\nabla^2 \phi(u)} \geq \frac{1}{10\sqrt{n}}.$$

Therefore, by Lemma 5.7.4,

$$d_P(u, v) \geq \frac{1}{10\sqrt{mn}}.$$

We can now bound the conductance of $S_1$. We may assume that $\text{vol}(S_i') \geq \text{vol}(S_i)/2$; otherwise, it immediately follows that the conductance of $S_1$ is at least $1/2000$. Assuming this,

$$\begin{aligned}
\int_{S_1} P_x(S_2)\, dx &\geq \frac{1}{2} \int_{S_3'} \frac{1}{8} dx \\
&\geq \frac{1}{16} \text{vol}(S_3') \\
\text{using isoperimetry} &\geq \frac{1}{16} d_P(S_1', S_2') \frac{\text{vol}(S_1')\text{vol}(S_2')}{\text{vol}(P)} \\
&\geq \frac{1}{1000\sqrt{mn}} \min\left\{\text{vol}(S_1), \text{vol}(S_2)\right\}.
\end{aligned}$$

□

## 5.8 Hamiltonian Monte Carlo

Hamiltonian dynamics is an alternative way to formulate Newtonian mechanics. The Hamiltonian $H$ captures both the potential and kinetic energy of a particle as a function of its position and velocity. The dynamics can be described by the following differential equations.

$$\begin{aligned}
\frac{dx}{dt} &= \frac{\partial H(x, v)}{\partial v}, \\
\frac{dv}{dt} &= -\frac{\partial H(x, v)}{\partial x}.
\end{aligned}$$

These equations preserve the Hamiltonian function $H$. In the simplest Euclidean setting, it can be defined as follows.

$$H(x, v) = f(x) + \frac{1}{2} \|v\|^2$$

so that

$$\frac{dx}{dt} = v, \quad \frac{dv}{dt} = -\nabla f(x)$$

or

$$\frac{d^2 x}{dt^2} = -\nabla f(x).$$

More generally, the Hamiltonian can depend on a function that defines a local metric:

$$H(x, v) = f(x) + \frac{1}{2} \log((2\pi)^n \det g(x)) + \frac{1}{2} v^T g(x)^{-1} v$$

where $g(x)$ is a matrix, and when it is PSD, it defines a local norm at $x$. In this sense, we can view the dynamics as evolving on a manifold with local metric $g(x)$. Here in this chapter, we will focus on the case when $g(x) = I$, the standard Euclidean metric.

(Riemannian) Hamiltonian Monte Carlo (or RHMC) [?, ?][?, ?] is a Markov Chain Monte Carlo method for sampling from a desired distribution. Each step of the method consists of the following: At a current point $x$,

1. Pick a random velocity $y$ according to a local distribution defined by $x$ (in the simplest setting, this is the standard Gaussian distribution for every $x$).
2. Move along the Hamiltonian curve defined by Hamiltonian dynamics at $(x, y)$ for time (distance) $\delta$.

For the choice of $H$ above, the marginal distribution of the current point $x$ approaches the target distribution with density proportional to $e^{-f}$. Note that HMC does not require a Metropolis filter! Thus, unlike the walks we have seen so far, its step sizes are not limited by this consideration even in high dimension. Hamiltonian Monte Carlo can be used for sampling from a general distribution $e^{-H(x,y)}$.

**Definition 5.8.1.** Given a continuous, twice-differentiable function $H : \mathcal{M} \times \mathbb{R}^n \subset \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ (*Hamiltonian*) where $\mathcal{M}$ is the $x$ domain of $H$, we say $(x(t), y(t))$ follows a *Hamiltonian curve* if it satisfies the *Hamiltonian equations*

$$\begin{aligned}\frac{dx}{dt} &= \frac{\partial H(x,y)}{\partial y},\\ \frac{dy}{dt} &= -\frac{\partial H(x,y)}{\partial x}.\end{aligned} \tag{5.8.1}$$

We define the map $T_\delta(x, y) \stackrel{\text{def}}{=} (x(\delta), y(\delta))$ where the $(x(t), y(t))$ follows the Hamiltonian curve with the initial condition $(x(0), y(0)) = (x, y)$.

Hamiltonian Monte Carlo is a Markov chain generated by a sequence of randomly Hamiltonian curves.

**Algorithm 18:** Hamiltonian Monte Carlo

```
Input: some initial point $x^{(1)} \in \mathcal{M}$.
for $i = 1, 2, \cdots, T$ do
    Sample $y$ according to $e^{-H(x^{(k)},y)}/\pi(x^{(k)})$ where $\pi(x) = \int_{\mathbb{R}^n} e^{-H(x,y)}dy$.
    With probability $\frac{1}{2}$, set $(x^{(k+1)}, y^{(k+1)}) = T_\delta(x^{(k)}, y)$.
    Otherwise, $(x^{(k+1)}, y^{(k+1)}) = T_{-\delta}(x^{(k)}, y)$.
end
Output: $(x^{(T+1)}, y^{(T+1)})$.
```

## Time-reversibility

**Lemma 5.8.2** (Energy Conservation)**.** *For any Hamiltonian curve $(x(t), y(t))$, we have that*

$$\frac{d}{dt}H(x(t), y(t)) = 0.$$

*Proof.* Note that

$$\frac{d}{dt}H(x(t), y(t)) = \frac{\partial H}{\partial x}\frac{dx}{dt} + \frac{\partial H}{\partial y}\frac{dy}{dt} = \frac{\partial H}{\partial x}\frac{\partial H}{\partial y} - \frac{\partial H}{\partial y}\frac{\partial H}{\partial x} = 0.$$

□

**Lemma 5.8.3** (Measure Preservation)**.** *For any $t \geq 0$, we have that*

$$\det\left(DT_t(x, y)\right) = 1$$

*where $DT_t(x, y)$ is the Jacobian of the map $T_t$ at the point $(x, y)$.*

*Proof.* Let $(x(t, s), y(t, s))$ be a family of Hamiltonian curves given by $T_t(x + sd_x, y + sd_y)$. We write

$$u(t) = \frac{\partial}{\partial s}x(t, s)|_{s=0}\,,\ v(t) = \frac{\partial}{\partial s}y(t, s)|_{s=0}.$$

By differentiating the Hamiltonian equations (5.8.1) w.r.t. $s$, we have that

$$\begin{aligned}\frac{du}{dt} &= \frac{\partial^2 H(x,y)}{\partial y \partial x}u + \frac{\partial^2 H(x,y)}{\partial y \partial y}v,\\ \frac{dv}{dt} &= -\frac{\partial^2 H(x,y)}{\partial x \partial x}u - \frac{\partial^2 H(x,y)}{\partial x \partial y}v,\\ (u(0),v(0)) &= (d_x,d_y).\end{aligned}$$

This can be captured by the following matrix ODE

$$\begin{aligned}\frac{d\Phi}{dt} &= \begin{pmatrix} \frac{\partial^2 H(x(t),y(t))}{\partial y \partial x} & \frac{\partial^2 H(x(t),y(t))}{\partial y \partial y} \\ -\frac{\partial^2 H(x(t),y(t))}{\partial x \partial x} & -\frac{\partial^2 H(x(t),y(t))}{\partial x \partial y}\end{pmatrix}\Phi(t)\\ \Phi(0) &= I\end{aligned}$$

using the equation

$$DT_t(x,y)\begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} u(t) \\ v(t) \end{pmatrix} = \Phi(t)\begin{pmatrix} d_x \\ d_y \end{pmatrix}.$$

Therefore, $DT_t(x,y) = \Phi(t)$. Next, we observe that

$$\frac{d}{dt}\log\det\Phi(t) = \mathrm{tr}\left(\Phi(t)^{-1}\frac{d}{dt}\Phi(t)\right) = \mathrm{tr}\begin{pmatrix} \frac{\partial^2 H(x(t),y(t))}{\partial y \partial x} & \frac{\partial^2 H(x(t),y(t))}{\partial y \partial y} \\ -\frac{\partial^2 H(x(t),y(t))}{\partial x \partial x} & -\frac{\partial^2 H(x(t),y(t))}{\partial x \partial y}\end{pmatrix} = 0.$$

Hence,

$$\det\Phi(t) = \det\Phi(0) = 1.$$

□

Using the previous two lemmas, we can see that Hamiltonian Monte Carlo indeed converges to the desired distribution.

**Lemma 5.8.4** (Time reversibility)**.** **Let $p_x(x')$ denote the probability density of one step of the Hamiltonian Monte Carlo starting at $x$. We have that*

$$\pi(x)p_x(x') = \pi(x')p_{x'}(x)$$

*for almost everywhere in $x$ and $x'$ where $\pi(x) = \int_{\mathbb{R}^n} e^{-H(x,y)}dy$.*

*Proof.* Fix $x$ and $x'$. Let $F_\delta^x(y)$ be the $x$ component of $T_\delta(x,y)$. Let $V_+ = \{y : F_\delta^x(y) = x'\}$ and $V_- = \{y : F_{-\delta}^x(y) = x')\}$. Then,

$$\pi(x)p_x(x') = \frac{1}{2}\int_{y\in V_+}\frac{e^{-H(x,y)}}{|\det(DF_\delta^x(y))|} + \frac{1}{2}\int_{y\in V_-}\frac{e^{-H(x,y)}}{\left|\det\left(DF_{-\delta}^x(y)\right)\right|}.$$

We note that this formula assumed that $DF_\delta^x$ is invertible. Sard's theorem showed that $F_\delta^x(N)$ is measure 0 where $N \overset{\text{def}}{=} \{y : DF_s^x(y) \text{ is not invertible}\}$. Therefore, the formula is correct except for a measure zero subset.

By reversing time for the Hamiltonian curve, we have that for the same $V_\pm$,

$$\pi(x')p_{x'}(x) = \frac{1}{2}\int_{y\in V_+}\frac{e^{-H(x',y')}}{\left|\det\left(DF_{-\delta}^{x'}(y')\right)\right|} + \frac{1}{2}\int_{y\in V_-}\frac{e^{-H(x',y')}}{\left|\det\left(DF_\delta^{x'}(y')\right)\right|} \tag{5.8.2}$$

where $y'$ denotes the $y$ component of $T_\delta(x,y)$ and $T_{-\delta}(x,y)$ in the first and second sum respectively.

We compare the first terms in both equations. Let $DT_\delta(x,y) = \begin{pmatrix} A & B \\ C & D\end{pmatrix}$. Since $T_\delta \circ T_{-\delta} = I$ and $T_\delta(x,y) = (x',y')$, the inverse function theorem shows that $DT_{-\delta}(x',y')$ is the inverse map of $DT_\delta(x,y)$. Hence, we have that

$$DT_{-\delta}(x',y') = \begin{pmatrix} A & B \\ C & D\end{pmatrix}^{-1} = \begin{pmatrix} \cdots & -A^{-1}B(D-CA^{-1}B)^{-1} \\ \cdots & \cdots\end{pmatrix}.$$

Therefore, we have that $F_\delta^x(y) = B$ and $F_{-\delta}^{x'}(y') = -A^{-1}B(D - CA^{-1}B)^{-1}$. Hence, we have that

$$\left|\det\left(DF_{-\delta}^{x'}(y')\right)\right| = \left|\det A^{-1}\det B\det\left(D - CA^{-1}B\right)^{-1}\right| = \frac{|\det B|}{\left|\det\begin{pmatrix} A & B \\ C & D\end{pmatrix}\right|}.$$

Using that $\det\left(DT_t(x,y)\right) = \det\begin{pmatrix} A & B \\ C & D\end{pmatrix} = 1$ (Lemma 5.8.3), we have that

$$\left|\det\left(DF_{-\delta}^{x'}(y')\right)\right| = \left|\det\left(DF_\delta^x(y)\right)\right|.$$

Hence, we have that

$$\begin{aligned}\frac{1}{2}\int_{y\in V_+}\frac{e^{-H(x,y)}}{\left|\det\left(DF_\delta^x(y)\right)\right|} &= \frac{1}{2}\int_{y\in V_+}\frac{e^{-H(x,y)}}{\left|\det\left(DF_{-\delta}^{x'}(y')\right)\right|}\\ &= \frac{1}{2}\int_{y\in V_+}\frac{e^{-H(x',y')}}{\left|\det\left(DF_{-\delta}^{x'}(y')\right)\right|}\end{aligned}$$

where we used that $e^{-H(x,y)} = e^{-H(x',y')}$ (Lemma 5.8.2) at the end.

For the second term in (5.8.2), by the same calculation, we have that

$$\frac{1}{2}\int_{y\in V_-}\frac{e^{-H(x,y)}}{\left|\det\left(DF_{-\delta}^{x}(y)\right)\right|} = \frac{1}{2}\int_{y\in V_+}\frac{e^{-H(x',y')}}{\left|\det\left(DF_{\delta}^{x'}(y')\right)\right|}$$

□

## Convergence

First we consider the convergence in the case when $H(x,v) = f(x) + \frac{1}{2}\|v\|^2$ for a strongly convex function $f$. So the marginal of the stationary distribution along $x$ is proportional to $e^{-f}$. The idea here is coupling (as we did for Langevin dynamics). We consider two separate processes $x$ and $y$, with their next step directions chosen to be identical. The key lemma is that with this setting the squared distance decreases up to a certain time that depends on the condition number.

**Lemma 5.8.5.** *Let $x(t)$ and $y(t)$ be the solution of HMC dynamics starting at $x(0)$ and $y(0)$ with initial direction $x'(0) = y'(0)$. Suppose that $f$ satisfies*

$$\mu\cdot I \preceq \nabla^2 f(x) \preceq L\cdot I$$

*for all $x$. Then, for $0 \le t \le \frac{\mu^{1/4}}{2L^{3/4}}$, we have that*

$$\|x(t) - y(t)\|_2^2 \le \left(1 - \frac{\mu}{4}t^2\right)\|x(0) - y(0)\|_2^2.$$

*With $T$ set to the upper bound above, we have*

$$\|x(T) - y(T)\|_2^2 \le \left(1 - \frac{1}{16}\left(\frac{\mu}{L}\right)^{\frac{3}{2}}\right)\|x(0) - y(0)\|_2^2.$$

This gives us the following bound, by induction.

**Theorem 5.8.6.** *Suppose $f$ satisfies*

$$\mu\cdot I \preceq \nabla^2 f(x) \preceq L\cdot I.$$

*For step-size $\delta = \frac{\mu^{1/4}}{2L^{3/4}}$ , let $X^T$ be and HMC sequence and $Y \sim e^{-f}$. Then, after $T = O\left(\left(\frac{L}{\mu}\right)^{\frac{3}{2}}\log\frac{n}{\mu\epsilon}\right)$ steps, we have $W_2(X,Y) \le \epsilon$.*

*Proof.* Besides the one-step contraction above, we need a bound on the initial distance. Let $x^{(\min)}$ be the minimum of $f$. Then, we have

$$\|x^{(0)} - y^{(0)}\|_2^2 \le 2\|x^{(0)} - x^{(\min)}\|_2^2 + 2\|y^{(0)} - x^{(\min)}\|_2^2.$$

The strong convexity of $f$ shows that

$$\|x^{(0)} - x^{(\min)}\|_2^2 \le \frac{1}{\mu^2}\|\nabla f(x^{(0)})\|_2^2.$$

For the second term, we use the following bound

$$\mathbf{E}\left[\|y^{(0)} - x^{(\min)}\|_2^2\right] \le \frac{d}{\mu}.$$

□

We now turn to the proof of the contraction bound. The following simple ODE bound will be useful for proving the main convergence lemma.

**Lemma 5.8.7.** *Given a continuous function $v(t)$ and positive scalars $\gamma$ such that*

$$0 \le v(t) \le \beta + \gamma \int_0^t (t-s)v(s)\,ds,$$

*then we have that $v(t) \le \beta\cosh(\sqrt{\gamma}t)$ for all $t \ge 0$.*

**Exercise 5.8.8.** Prove the above lemma. [Hint: first do it assuming the upper bound for each $t$ is equality.]

*Proof of Lemma 5.8.5.* Let the error function be $u(t) = y(t) - x(t)$. The definition of HMC shows that

$$u''(t) = -(\nabla f(y(t)) - \nabla f(x(t))) = -H(t)\cdot u(t)$$

where

$$H(t) = \int_0^1 \nabla^2 f(x(t) + s(y(t) - x(t)))\,ds$$

We have that $\mu \cdot I \preceq H(t) \preceq L \cdot I$.

Without loss of generality, we can assume $\|u(0)\|_2 = 1$ and $u(0) = e_1$. We first get a rough bound on $\|u(t)\|_2$ saying that it does not change by more than a constant factor, and then use it to get a strong bound showing contraction.

*Proof.* Integrating both sides of $u''(t) = -H(t)\cdot u(t)$ twice and using $u'(0) = 0$ gives

$$u(t) = u(0) - \int_0^t (t-s)H(s)u(s)\,ds.$$

Therefore,

$$\|u(t)\|_2 \le 1 + L\cdot\int_0^t (t-s)\|u(s)\|_2\,ds.$$

Applying Lemma 5.8.7 to this equation with $t \le \frac{1}{2\sqrt{L}}$ gives

$$\|u(t)\|_2 \le \cosh(\sqrt{L}t) \le \frac{6}{5}.$$

Putting it back in the integral equation,

$$\|u(t) - e_1\|_2 \le \int_0^t (t-s)\|H(s)\cdot u(s)\|_2\,ds = \int_0^t (t-s)\cdot L\cdot\frac{6}{5}\,ds = \frac{6}{10}Lt^2.$$

In particular, for any $0 \le t \le \frac{1}{2\sqrt{L}}$, we have that

$$\frac{5}{6} \le u_1(t) \le \frac{7}{6}.$$

Let $P_1$ be the orthogonal projection to the first coordinate and $P_{-1} = I - P_1$. We write $u(t) = u_1(t) + u_{-1}(t)$ with $u_1(t) = P_1 u(t)$ and $u_{-1}(t) = P_{-1}u(t)$, namely, $u_1(t)$ is parallel to $e_1$ and $u_{-1}(t)$ is orthogonal to $e_1$.

Fix any $0 \leq t \leq T$. Let $\beta(t) = e_1^\top u(t)$. By the definition of $u$, we have $u''(t) = -\beta(t) \cdot H(t)e_1 - H(t)u_{-1}(t)$. Integrating both sides twice and using $u_{-1}(0) = 0$, we have

$$\begin{aligned} u_{-1}(t) &= \int_0^t (t-s)P_{-1}u''(s)\,ds \\ &= -\int_0^t (t-s)\cdot\beta(s)\cdot P_{-1}H(s)e_1\,ds - \int_0^t (t-s)P_{-1}H(s)u_{-1}(s)\,ds. \end{aligned}$$

Using $0 \preceq H(t) \preceq L\cdot I$ and $\frac{5}{6} \leq \beta(s) \leq \frac{7}{6}$, we have that

$$\begin{aligned} \|u_{-1}(t)\|_2 &\leq \int_0^t (t-s)\cdot\beta(s)\cdot\|H(s)e_1\|_2\,ds + L\cdot\int_0^t (t-s)\cdot\|u_{-1}(s)\|_2\,ds \\ &\leq \frac{7}{6}\int_0^t (t-s)\cdot\|H(s)e_1\|_2\,ds + L\cdot\int_0^t (t-s)\cdot\|u_{-1}(s)\|_2\,ds \\ &\leq \frac{7}{6}\alpha(T) + L\cdot\int_0^t (t-s)\cdot\|u_{-1}(s)\|_2\,ds \end{aligned} \tag{5.8.3}$$

where $\alpha(T) \overset{\text{def}}{=} \int_0^T (T-s)\cdot\|H(s)e_1\|_2\,ds$. Solving this integral inequality using Lemma 5.8.7, we get

$$\|u_{-1}(t)\|_2 \leq \frac{7}{6}\alpha(T)\cdot\cosh(\sqrt{L}t) \leq \frac{7}{6}\alpha(T)\cdot\cosh(1/2) \leq \frac{4}{3}\alpha(T) \tag{5.8.4}$$

where the second step follows from $t \leq \frac{1}{2\sqrt{L}}$, and the last step follows from $\cosh(1/2) \leq \frac{8}{7}$.

Next we look at the first coordinate.

$$\begin{aligned} \beta''(t) &= -\beta(t)\cdot e_1^\top H(t)e_1 - e_1^\top H(t)u_{-1}(t) \\ &\leq -\frac{5}{6}e_1^\top H(t)e_1 + \|H(t)e_1\|_2\cdot\|u_{-1}(t)\|_2. \end{aligned} \tag{5.8.5}$$

To bound the last term, we note that

$$\begin{aligned} \|H(t)e_1\|_2 &= \sqrt{e_1^\top H^2(t)e_1} \\ &\leq \sqrt{L\cdot e_1^\top H(t)e_1} && \text{since } H^2(t) \preceq L\cdot H(t) \\ &\sqrt{\frac{L}{\mu}}\sqrt{\frac{\mu}{e_1^\top H(t)e_1}}e_1^\top H(t)e_1 \\ &\leq \sqrt{\frac{L}{\mu}}\cdot e_1^\top H(t)e_1 && \text{since } \mu \leq e_1^\top H(t)e_1. \end{aligned}$$

We bound $\alpha(t)$ as follows

$$\alpha(t) = \frac{1}{\|e(0)\|_2}\int_0^t (t-s)\cdot\|H(s)e(0)\|_2\,ds \leq L\int_0^t (t-s)\,ds = L\frac{t^2}{2}.$$

Therefore, for $0 \leq t \leq \frac{\mu^{1/4}}{2L^{3/4}}$, we have that $\alpha(T) \leq \frac{1}{8}\sqrt{\frac{\mu}{L}}$. Using these bounds,

$$\beta''(t) \leq -e_1^\top H(t)e_1\cdot\left(\frac{5}{6} - \frac{4}{3}\sqrt{\frac{L}{\mu}}\alpha(T)\right) \leq -\frac{2}{3}e_1^\top H(t)e_1.$$

Hence,

$$\beta(t) \leq 1 - \frac{2}{3}\int_0^t (t-s)\cdot e_1^\top H(s)e_1\,ds. \tag{5.8.6}$$

Using this, $\beta(t) \geq \frac{5}{6}$ and (5.8.4),

$$\begin{aligned}
\|u(t)\|_2^2 &= \beta^2(t) + \|u_{-1}(t)\|_2^2 \\
&\leq 1 - \int_0^t (t-s) \cdot e_1^\top H(s) e_1 \, ds + \left( \frac{4}{3} \int_0^T (T-s) \cdot \|H(s)e_1\|_2 \, ds \right)^2 \\
&\leq 1 - \int_0^t (t-s) \cdot e_1^\top H(s) e_1 \, ds + 2\sqrt{\frac{L}{\mu}} \int_0^T (T-s) \cdot e_1^\top H(s) e_1 \, ds \cdot \alpha(T) \\
&= 1 - \int_0^t (t-s) \cdot e_1^\top H(s) e_1 \, ds \left( 1 - 2\sqrt{\frac{L}{\mu}} \alpha(T) \right) \\
&\leq 1 - \frac{1}{2} \int_0^t (t-s) \cdot e_1^\top H(s) e_1 \, ds
\end{aligned}$$

where we used the bound on $\|H(t)e_1\|_2$at the second step and $\alpha(T) \leq \frac{1}{8}\sqrt{\frac{\mu}{L}}$ at the end.

Now we can bound $e_1^\top H(s) e_1 \geq L$ and get

$$\int_0^t (t-s) \cdot e_1^\top H(s) e_1 \, ds \geq \frac{\mu}{2} t^2$$

which implies

$$\|u(t)\|_2^2 \leq 1 - \frac{\mu}{4} t^2.$$

□

□

# Chapter 6

# Sparsification

In this chapter, we study various sampling/randomization techniques for convex optimization.

## 6.1 Subspace embedding

In this section and next section, we consider the least squares regression problem

$$\min_x \|Ax - b\|_2^2$$

where $A \in \mathbb{R}^{n \times d}$ with $n \gg d$. The gradient of the function is $2A^\top Ax - 2A^\top b$. Therefore, the solution is given by

$$x = (A^\top A)^{-1} A^\top b.$$

If the matrix $A^\top A \in \mathbb{R}^{d \times d}$ is given, then, we can solve the equation above in time $d^\omega$. If $n > d^\omega$, then the bottleneck is simply to compute $A^\top A$. The following lemma shows that it suffices to approximate $A^\top A$. We will use the following general norm. For a vector $x$, $\|x\|_A = \sqrt{x^T A x}$; for a matrix $B$,

$$\|B\|_A = \sup_x \frac{\|Bx\|_A}{\|x\|_A}.$$

**Exercise 6.1.1.** Show that if $\|B\|_A \leq \alpha$ then $B^T A B \preceq \alpha A$.

The simplest iteration is the Richardson iteration:

$$x^{(k+1)} = b + (I - A^T A) x^{(k)} = x^{(k)} + b - A^T A x^{(k)}.$$

To ensure this converges, we scale down $A^T A$ by its largest eigenvalue so that $A^T A \prec I$. This gives a bound of $O(\kappa(A^T A) \log(1/\epsilon))$ on the number of iterations to get $\epsilon$ error. More generally, one can use *pre-conditioning.*

**Lemma 6.1.2.** *Given a matrix $M$ such that $A^\top A \preceq M \preceq \kappa \cdot A^\top A$ for some $\kappa \geq 1$. Consider the algorithm $x^{(k+1)} = x^{(k)} - M^{-1}(A^\top A x^{(k)} - A^\top b)$ . Then, we have that*

$$\|r^{(k)}\|_{(A^\top A)^{-1}} \leq \left(1 - \frac{1}{\kappa}\right)^k \|r^{(0)}\|_{(A^\top A)^{-1}}.$$

*where $r^{(k)} = A^\top A x^{(k)} - A^\top b$.*

*Remark* 6.1.3. The proof also shows why the choice of norm above is the natural one. In this norm, the residual drops geometrically.

*Proof.* Expanding the residual term $r^{(k+1)}$ gives

$$\begin{aligned} r^{(k+1)} &= A^\top A x^{(k+1)} - A^\top b \\ &= A^\top A x^{(k+1)} - A^\top b - A^\top A M^{-1}(A^\top A x^{(k)} - A^\top b) \\ &= (I - A^\top A M^{-1}) r^{(k)}. \end{aligned}$$

Therefore, the norm of residual is

$$\|r^{(k+1)}\|^2_{(A^\top A)^{-1}} = r^{(k)\top}(I - M^{-1}A^\top A)(A^\top A)^{-1}(I - A^\top AM^{-1})r^{(k)}.$$

The matrix in the middle can be upper bounded as follows

$$\begin{aligned}&(I - M^{-1}A^\top A)(A^\top A)^{-1}(I - A^\top AM^{-1})\\ =&(A^\top A)^{-1} - 2M^{-1} + M^{-1}A^\top AM^{-1}\\ =&(A^\top A)^{-\frac{1}{2}}(I-H)^2(A^\top A)^{-\frac{1}{2}}\end{aligned}$$

where $H = (A^\top A)^{\frac{1}{2}}M^{-1}(A^\top A)^{\frac{1}{2}}$. Note that the eigenvalues of $H$ lie between $1/\kappa$ and 1 and hence

$$\lambda_{\max}(I-H)^2 \leq \left(1 - \frac{1}{\kappa}\right)2.$$

This gives the conclusion. □

Note that $x^\top A^\top Ax = \|Ax\|^2$. Alternatively, we can think our goal is to approximate $A$ by $B$ s.t. $\|Ax\|^2$ is close to$\|Bx\|^2$ for all $x$. In this section, we show that we can simply take $B = \Pi A$ for a random matrix $\Pi$. With this choice of $M$, we can run the Richardson iteration. We need to see if this will make the entire procedure more efficient.

**Definition 6.1.4.** A matrix $\Pi \in \mathbb{R}^{m\times n}$ is an embedding for a set $S \subset \mathbb{R}^n$ with distortion $\epsilon$ if

$$(1-\epsilon)\|y\|^2 \leq \|\Pi y\|^2 \leq (1+\epsilon)\|y\|^2$$

for all $y \in S$.

In this section, we focus on the case that $S$ is a $d$-dimensional subspace in $\mathbb{R}^n$, namely $S = \{Ax : x \in \mathbb{R}^d\}$. Consider the SVD $A = U\Sigma V^\top$. For any $y \in S$, we have that

$$\|U^\top y\|_2^2 = \|U^\top U\Sigma V^\top x\|_2^2 = x^\top V\Sigma V^\top x = \|Ax\|^2.$$

Therefore, any $d$-dimensional subspace has a zero distortion embedding of size $d \times n$.

**Exercise 6.1.5.** For any $d$-dimensional subspace $S$, any embedding with distortion $\epsilon < 1$ must have at least $d$ rows.

This embedding is not useful for solving the least squares problem because the solution of the least square problem is simply a close form of the SVD decomposition $x = V\Sigma^{-1}U^\top b$ and that finding SVD is usually strictly more expensive.

## Oblivious Subspace Embedding via Johnson-Lindenstrauss

Surprisingly, there are random embeddings that have a small distortion without knowledge of the subspace.

**Definition 6.1.6.** A random matrix $\Pi \in \mathbb{R}^{m\times d}$ is a $(d,\epsilon,\delta)$-oblivious subspace embedding (OSE) if for any fixed $d$-dimensional subspace $S$, $\Pi$ is an embedding for $S$ with distortion $\epsilon$ with probability at least $1-\delta$.

The next lemma provides an equivalent definition.

**Lemma 6.1.7.** *We call $\Pi$ is a $(d,\epsilon,\delta)$ OSE if for any matrix $U \in \mathbb{R}^{n\times d}$ with orthonormal columns, we have that*

$$\mathbf{P}(\|U^\top\Pi^\top\Pi U - I_d\|_{\text{op}} \leq \epsilon) \geq \delta.$$

*Proof.* Let $S$ be the subspace with an orthonormal basis $U \in \mathbb{R}^{n\times d}$, namely $S = \{y : Uz\}$ Then, the condition

$$(1-\epsilon)\|y\|^2 \leq \|\Pi y\|^2 \leq (1+\epsilon)\|y\|^2$$

can be rewritten as

$$(1-\epsilon)I_d \preceq U^\top\Pi^\top\Pi U \preceq (1+\epsilon)I_d.$$

□

For the special case $d = 1$, the definition becomes

$$\mathbf{P}\left[\left|\|\Pi a\|_2^2 - 1\right| \le \epsilon\right] \ge 1 - \delta$$

for all unit vectors $a$. An OSE for $d = 1$ is given by the Johnson-Lindenstrauss Lemma. The original version was for a uniform random subspace of dimension $d$, but later versions extended this extended this to Gaussian, Bermoulli and more general random matrices [62].

**Lemma 6.1.8** (Johnson-Lindenstrauss Lemma)**.** *Let $\Pi \in \mathbb{R}^{m\times d}$ be a random matrix with i.i.d entries from $N(0, \frac{1}{\sqrt{m}})$ or uniformly sampled from $\pm\frac{1}{\sqrt{m}}$ with $m = \Theta(\frac{1}{\epsilon^2}\log(\frac{1}{\delta}))$. Then, $\Pi$ is a $(1, \epsilon, \delta)$ OSE.*

We will skip the proof for this as we will prove a more general result later. Next, we show that any OSE for $d = 1$ is a OSE for general $d$. Therefore, it suffices to focus on the case $d = 1$. First, we need a lemma about $\epsilon$-net on $S^{n-1}$.

**Lemma 6.1.9.** *For any $\epsilon > 0$ and any $n \in \mathbb{N}$, there are $(1+\frac{2}{\epsilon})^n$ unit vectors $x_i \in \mathbb{R}^n$ such that for any unit vector $x \in \mathbb{R}^n$, there is $i$ such that $\|x - x_i\|_2 \le \epsilon$.*

*Remark.* We call the points $\{x_i\}_i$ the $\epsilon$-net on $S^{n-1}$.

*Proof.* We consider the following algorithm:

1. $V = S^{n-1}$.
2. For $i = 1, \cdots,$
   (a) Pick any $x_i \in V$. (If there is no such $x_i$, end.)
   (b) $V \leftarrow V \backslash B(x_i, \epsilon)$.

Let $\{x_1, x_2, \cdots, x_N\}$ be the points it found. By the construction, it is clear that $\|x - x_i\| \le \epsilon$ for some $i$ (else the procedure would continue). Now consider balls of radius $\epsilon/2$ centered at each $x_i$. To bound the number of points, we note that all $B(x_i, \frac{\epsilon}{2})$ are disjoint and that all balls lie in $B(0, 1+\frac{\epsilon}{2})$. Therefore,

$$N \cdot \mathrm{vol}(B(0, \frac{\epsilon}{2})) \le \mathrm{vol}(B(0, 1 + \frac{\epsilon}{2})).$$

and hence $N \le (1+\frac{2}{\epsilon})^n$. □

**Lemma 6.1.10.** *Suppose that $\Pi$ is a $(1, \epsilon, \delta)$ OSE. Then, $\Pi$ is a $(d, 4\epsilon, 5^d\delta)$ OSE.*

*Proof.* Let $N = \{x_i\}_{i=1}$ are the $\frac{1}{2}$-net for $S^{d-1}$. Then, for any $x$, we have $x_1 \in N$ such that $\|x - x_{i_1}\|_2 \le \frac{1}{2}$. Using the $\frac{1}{2}$-net guarantee on the vector $x - x_1$, we can find $x_2 \in N$ and $0 \le t_2 \le \frac{1}{2}$ such that

$$\|x - x_1 - t_2 x_2\| \le \frac{1}{4}.$$

Similarly, we have $x = \sum_{i=1}^{\infty} t_i x_i$ with $0 \le t_i \le \frac{1}{2^i}$. Hence, we have that

$$\begin{aligned} x^\top(U^\top\Pi^\top\Pi U - I_d)x &= \sum_{i,j} t_i t_j x_i^\top (U^\top\Pi^\top\Pi U - I_d) x_j \\ &\le \sum_{i,j} t_i t_j \cdot \max_{x\in N}\left|x^\top(U^\top\Pi^\top\Pi U - I_d)x\right| \\ &= 4\max_{x\in N}\left|x^\top(U^\top\Pi^\top\Pi U - I_d)x\right| \\ &= 4\max_{x\in UN}\left|\|\Pi x\|^2 - 1\right| \end{aligned}$$

Now, using that $\Pi$ is $(1, \epsilon, \delta)$ OSE, we have that

$$\mathbf{P}(\left|\|\Pi x\|^2 - 1\right| \le \epsilon) \ge 1 - \delta.$$

Take a union bound over all $x \in UN$, we have

$$\mathbf{P}(x^\top(U^\top \Pi^\top \Pi U - I_d)x \leq 4\epsilon) \geq 1 - 5^d \delta.$$

□

This reduction and the Johnson-Lindenstrauss Lemma shows that a random $\pm\frac{1}{\sqrt{m}}$ is a $(d, \epsilon, \delta)$ OSE with

$$m = \Theta(\frac{1}{\epsilon^2}(d + \log(\frac{1}{\delta}))).$$

As we discussed before that any $(d, \epsilon, \delta)$ OSE should have at least $d$ rows. Therefore, the number of rows of this OSE is tight for the the regime $\epsilon = \Theta(1)$. We only need an OSE for $\epsilon = \Theta(1)$ because of Lemma 6.1.2; by iterating we can get any $\epsilon$ with an overhead of $\log(1/\epsilon)$. Unfortunately, computing $\Pi A$ is in fact more expensive than $A^\top A$. The first involves multiplying $\Theta(d) \times n$ and $n \times d$ matrix, the second one involves multiplying $d \times n$ and $n \times d$ matrix.

## Sparse JL*

We consider the sparse matrix $\Pi \in \mathbb{R}^{m \times n}$ where each entry is $\pm\frac{1}{\sqrt{s}}$ with probability $\frac{s}{m}$ and 0 otherwise. We will show that this random matrix is a $(d, \epsilon, \delta)$ OSE for $s = \Theta(\frac{\log^2(d/\delta)}{\epsilon^2})$ and $m = \Theta(d \log(\frac{d}{\delta})/\epsilon^2)$. When $n = d = 1$, $\|\Pi x\|^2$ is exactly equals to how many non-zeros in the only column. Therefore, we indeed need $s$ scales like $1/\epsilon^2$. It turns out that one can select exactly $s$ non-zeros for each column. This allows us to use $s = \Theta(\frac{\log(d/\delta)}{\epsilon})$. The proof of this is a slightly more complicated due to the lack of independence [12].

To analyze $U^\top \Pi^\top \Pi U$, we note that

$$U^\top \Pi^\top \Pi U = \sum_r (\Pi U)_r^\top (\Pi U)_r.$$

Since each row of $\Pi$ is independent, we can simply use a matrix concentration bound to analyze the sum above. See [59] for a survey on matrix concentration bounds.

**Theorem 6.1.11** (Matrix Chernoff). *Given a sequence of independent random self-adjoint matrices $M_j \in \mathbb{R}^{n \times n}$ such that $\mathbf{E} M_j = I$ and $0 \preceq M_j \preceq R \cdot I$. Then, for $T = \Omega(\frac{R}{\varepsilon^2} \log \frac{n}{\delta})$, we have that*

$$(1 - \varepsilon) I \preceq \frac{1}{T} \sum_{j=1}^{T} M_j \preceq (1 + \varepsilon) I$$

*with probability $1 - \delta$.*

For our problem, we have $M_r = m \cdot U^\top \pi_r \pi_r^\top U$ where $\pi_r$ is the $r$-th row of $\Pi$. One can check that $X_r \succeq 0$ and that $\mathbf{E} X_r = I$. Next, we note that

$$X_r \preceq m \cdot \pi_r^\top U U^\top \pi_r \cdot I.$$

With small probability, $\pi_r^\top U U^\top \pi_r$ can be huge. However, as long as $\pi_r^\top U U^\top \pi_r$ is bounded by $R$ with probability $1 - \delta$, then we can still use the matrix Chernoff bound above. To bound $\pi_r^\top U U^\top \pi_r$, we will use the following large deviation inequality.

**Lemma 6.1.12** (Sparse Hanson-Wright Inequality). *For $\sigma_1, \cdots, \sigma_n$ are independent random variables with $\mathbf{E}\sigma_i = 0$, $|\sigma_i| \leq K$ and $\sigma_i \neq 0$ with probability $p$ for all $i$. Let $A \in \mathbb{R}^{n \times n}$, we have that*

$$\mathbf{P}(|\sigma^\top A \sigma - \mathbf{E}\sigma^\top A \sigma| > Kt) \leq \exp(-c \min(\frac{t^2}{p \sum_k A_{kk}^2 + p^2 \|A\|_F^2}, \frac{t}{\|A\|_{\mathrm{op}}})).$$

*for some universal constant $c$.*

Now, we can bound $R$.

**Lemma 6.1.13.** *If $s = \Omega(\frac{m}{d} \log(\frac{1}{\delta}))$, we have that $m \cdot \pi_r^\top U U^\top \pi_r \leq 2d$ with probability $1 - \delta$.*

*Proof.* Let $P = UU^\top$. The sparse Hanson-Wright inequality shows that

$$\mathbf{P}(|\sigma^\top P\sigma - \mathbf{E}\sigma^\top P\sigma| > \frac{t}{s}) \leq 2\exp(-c\min(\frac{t^2}{\frac{s}{m}\sum_k P_{kk}^2 + (\frac{s}{m})^2\|P\|_F^2}, \frac{t}{\|P\|_{\text{op}}})).$$

Note that $\mathbf{E}\pi_r^\top P\pi_r = \frac{1}{m}\text{tr}P$ and that $P$ is a projection matrix of rank $d$. Hence, we have $\text{tr}P = d$, $\sum_k P_{kk}^2 \leq d$ and $\|P\|_F = \sqrt{d}$ and $\|P\|_{\text{op}} = 1$. Hence, we have

$$\mathbf{P}(\sigma^\top P\sigma > \frac{t}{s} + \frac{d}{m}) \leq 2\exp(-c\min(\frac{t^2}{\frac{s}{m}d}, t)).$$

Setting $t = \frac{ds}{m}$, we have

$$\mathbf{P}(\sigma^\top P\sigma > 2\frac{d}{m}) \leq 2\exp(-c\frac{ds}{m}).$$

Setting $s = \Omega(\frac{m}{d}\log(\frac{1}{\delta}))$, we have that $\mathbf{P}(\sigma^\top P\sigma < 2\frac{d}{m}) \leq 1 - \delta$. □

Applying the matrix Chernoff bound, we have that if $m = \Omega(\frac{R}{\varepsilon^2}\log\frac{n}{\delta}) = \Omega(\frac{d}{\varepsilon^2}\log\frac{n}{\delta})$ and that $s = \Omega(\frac{m}{d}\log(\frac{1}{\delta}))$, then we have that $\Pi$ is a $(d, \epsilon, \delta)$ OSE.

## Back to Regression

In the last subsection, we presented a proof sketch of a suboptimal OSE. For completeness, we state a tighter version here:

**Theorem 6.1.14** ([12]). *Consider a random sparse matrix $\Pi \in \mathbb{R}^{m\times n}$ constructed by selecting exactly $s$ random entries in each column and assigning random $\pm\frac{1}{\sqrt{s}}$ independently. Suppose that $m = \Omega(\frac{d\log(\frac{d}{\delta})}{\epsilon^2})$ and $s = \Omega(\frac{\log(\frac{d}{\delta})}{\epsilon})$, then we have that $\Pi$ is $O(d, \epsilon, \delta)$ OSE.*

**Algorithm 19:** `RegressionUsingOSE`

```
Input: a matrix $A \in \mathbb{R}^{n\times d}$, a vector $b \in \mathbb{R}^d$, success probability $\delta$ and target accuracy $\epsilon$.
Let $\Pi$ be a $O(d, \frac{1}{4}, \delta)$ OSE constructed by Theorem 6.1.14.
Compute $A' = 2\Pi A$, $M = A'^\top A'$ and $M^{-1}$.
$x^{(1)} = 0$.
for $k = 1, 2, \cdots, 4\log(\frac{1}{\epsilon})$ do
    $x^{(k+1)} = x^{(k)} - M^{-1}(A^\top A x^{(k)} - A^\top b)$
end
return $x^{(\text{last})}$.
```

**Theorem 6.1.15.** *Given any matrix $A \in \mathbb{R}^{n\times d}$, any vector $b \in \mathbb{R}^d$, the algorithm* `RegressionUsingOSE` *returns a vector $x$ such that*

$$\|A^\top Ax - A^\top b\|_{(A^\top A)^{-1}} \leq \epsilon\|A^\top b\|_{(A^\top A)^{-1}}.$$

*with probability $1-\delta$ in $O(\text{nnz}(A)\log(\frac{d}{\delta\epsilon}) + d^\omega\log(\frac{d}{\delta}))$ time.*

*Proof.* The guarantee of $x$ follows from the definition of OSE and Lemma 6.1.2. We note that $\Pi A$ simply involves duplicating each row of $A$ into $s$ many rows in $\Pi A$. Hence, the cost of computing $\Pi A$ is $O(s\cdot\text{nnz}(A)) = O(\log(\frac{d}{\delta})\cdot\text{nnz}(A))$. The cost of computing $M$ takes $O(d^\omega\log(\frac{d}{\delta}))$ time. Computing $M^{-1}$ takes $O(d^\omega)$ time. The loop takes $O(\log(\frac{1}{\epsilon})\cdot\text{nnz}(A))$ time. This explain the total time. □

Linear regression can be solved in time $O(nnz(A)) + \widetilde{O}(d^\omega)$ via a very sparse embedding: each column of $\Pi$ picks exactly one nonzero entry in a random location. This was proved to work by Clarkson and Woodruff [64]. See also [28] for a survey.

## ■ 6.2 Leverage Score Sampling

Now, we give a way to reduce solving a tall linear regression system into a sequence of submatrices:

**Theorem 6.2.1** ([13]). *Given a matrix $A \in \mathbb{R}^{n\times d}$. Let $T(m)$ be the cost of solving $B^\top Bx = b$ among all $m$ distinct subrows $B$ of $A$. Then, we have that*

$$T(n) = O^*(\mathrm{nnz}(A) + T(O(d\log d))).$$

*Remark.* Here we use $O^*$ to emphasize there is some hidden dependence on $\log(\frac{1}{\epsilon})$ suppressed for notational simplicity. Lemma 6.1.2 shows that once we can solve the system with constant approximation, we can repeat $\log(\frac{1}{\epsilon})$ times to get an $\epsilon$-accurate solution.

### Leverage scores

The key concept in this reduction is leverage scores.

**Definition 6.2.2.** Given a matrix $A \in \mathbb{R}^{n\times d}$. Let $a_i^\top$ be the $i^{th}$ rows of $A$ and the leverage score of the $i^{th}$ row of $A$ is

$$\sigma_i(A) \stackrel{\text{def}}{=} a_i^\top (A^\top A)^+ a_i.$$

Note that $\sigma(A)$ is the diagonal of the projection matrix $A(A^\top A)^+A^\top$. Since $0 \preceq A(A^\top A)^+A^\top \preceq I$, we have that $0 \le \sigma_i(A) \le 1$. Furthermore, since $A(A^\top A)^+A^\top$ is a projection matrix, the sum of $A$'s leverage scores is equal to the matrix's rank:

$$\sum_{i=1}^n \sigma_i(A) = \mathrm{tr}(A(A^\top A)^+A^\top) = \mathrm{rank}(A(A^\top A)^+A^\top) = \mathrm{rank}(A) \le d. \tag{6.2.1}$$

A row's leverage score measures how important it is in composing the row space of $A$. If a row has a component orthogonal to all other rows, its leverage score is 1. Removing it would decrease the rank of $A$, completely changing its row space. The *coherence* of $A$ is $\|\sigma(A)\|_\infty$. If $A$ has low coherence, no particular row is especially important. If $A$ has high coherence, it contains at least one row whose removal would significantly affect the composition of $A$'s row space. Two characterizations that helps with this intuition follow.

**Lemma 6.2.3.** *For all $A \in \mathbb{R}^{n\times d}$ and $i \in [n]$ we have that*

$$\sigma_i(A) = \min_{A^\top x = a_i} \|x\|_2^2 .$$

*where $a_i$ is the $i^{th}$ row of $A$.*

**Lemma 6.2.4.** *For all $A \in \mathbb{R}^{n\times d}$ and $i \in [n]$, we have that $\sigma_i(A)$ is the smallest $t$ such that*

$$a_i a_i^\top \preceq t \cdot A^\top A. \tag{6.2.2}$$

Sampling rows from $A$ according to their exact leverage scores gives a spectral approximation for $A$ with high probability. Sampling by leverage score overestimates also suffices.

**Lemma 6.2.5.** *Given a vector $u$ of leverage score overestimates, i.e., $\sigma_i(A) \le u_i$ for all $i$. Let $X$ be the random matrix that is $\frac{1}{p_i}a_i a_i^\top$ with probability $p_i = \frac{u_i}{\|u\|_1}$. For $T = \Omega(\frac{\|u\|_1 \log n}{\varepsilon^2})$, with probability $1 - \frac{1}{n^{O(1)}}$, we have that*

$$(1-\varepsilon)A^\top A \preceq \frac{1}{T}\sum_{i=1}^T X_i \preceq (1+\varepsilon)A^\top A$$

*where $X_i$ are independent copies of $X$.*

*Proof.* Note that $\mathbf{E}X = A^\top A$ and that

$$0 \preceq X = \frac{1}{p_i}a_i a_i^\top \preceq \frac{\|u\|_1}{\sigma_i} a_i a_i^\top \preceq \|u\|_1 \cdot A^\top A.$$

Now, the statement simply follows from the matrix Chernoff bound with $M_k = (A^\top A)^{-\frac{1}{2}} X_k (A^\top A)^{-\frac{1}{2}}$ and $R = \|u\|_1$. □

Combining Lemma 6.2.5 and Lemma 6.1.2, we have that

$$T(n) = \text{cost of computing } \sigma_i + O^*(\text{nnz}(A) + T(d \log d)) \tag{6.2.3}$$

where we used that $\|\sigma\|_1 = O(d)$. However, computing $\sigma$ exactly is too expensive for many purposes. In [?], they showed that we can compute leverage scores, $\sigma$, approximately by solving only polylogarithmically many regression problems. This result uses the fact that

$$\sigma_i(A) = \left\|A(A^\top A)^+ A^\top e_i\right\|_2^2$$

and that by the Johnson-Lindenstrauss Lemma these lengths are persevered up to a multiplicative error if we project these vectors to random low dimensional subspaces.

In particular, this lemma shows that we can approximate $\sigma_i(A)$ via $\left\|\Pi A(A^\top A)^+ A^\top e_i\right\|_2^2$. The benefit of this is that we can compute $\Pi A(A^\top A)^+$ by solving $\frac{\log n}{\varepsilon^2}$ many linear systems. In other words, we have that

$$\text{cost of approximating } \sigma_i = O^*(\text{nnz}(A)) + O^*(T(n)).$$

Putting it into (6.2.3), we have this useless formula

$$T(n) = O^*(\text{nnz}(A) + T(n) + T(d \log d)).$$

This is a chicken and egg problem. To solve an overdetermined system faster, we want to use leverage score to sample the rows. And to approximate the leverage score, we need to solve the original overdetermined system. (Even worse, we need to solve it few times.)

## Uniform Sampling

The key idea to break this chicken and egg problem is to use uniform sampling. We define

$$\sigma_{i,S} = a_i^\top (A_S^\top A_S)^{-1} a_i$$

where $A_S$ is $A$ restricted to rows in $S$. Note that $A_S^\top A_S$ is an overestimate of $\sigma_i$. Hence, it suffices to bound $\|\sigma_{i,S}\|_1$. The key lemma is the following:

**Lemma 6.2.6.** *We have that*

$$\mathbf{E}_{|S|=k} \sum_{i=1}^n \sigma_{i,S\cup\{i\}} \leq \frac{nd}{k}.$$

*Proof.* Note that

$$\mathbf{E}_{|S|=k} \sum_{i=1}^n \sigma_{i,S\cup\{i\}} = \mathbf{E}_{|S|=k} \sum_{i\notin S} \sigma_{i,S\cup\{i\}} + \mathbf{E}_{|S|=k} \sum_{i\in S} \sigma_{i,S\cup\{i\}}.$$

Note that $\sum_{i\in S} \sigma_{i,S\cup\{i\}} = \sum_{i\in S} \sigma_{i,S} \leq d$. Hence, the second term is bounded by $n$.

For the first term, we note that "sample a set $S$ of size $k$, then sample $i \notin S$" is same as "sample a set $T$ of size $k+1$, then sample $i \in T$". Hence, we have

$$\begin{aligned}\mathbf{E}_{|S|=k} \mathbf{E}_{i\notin S} \sigma_{i,S\cup\{i\}} &= \mathbf{E}_{|T|=k+1} \mathbf{E}_{i\in T} \sigma_{i,T} \\ &\leq \mathbf{E}_{|T|=k+1} \frac{d}{k+1} = \frac{d}{k+1}\end{aligned}$$

Hence, we have that

$$\mathbf{E}_{|S|=k} \sum_{i=1}^n \sigma_{i,S\cup\{i\}} \leq \frac{d}{k+1}(n-k) + d = d \cdot \frac{n+1}{k+1}.$$

□

Next, using Sherman Morrison formula, we have that

$$\sigma_{i,S\cup\{i\}} = \begin{cases} \sigma_{i,S} & \text{if } i \in S \\ \frac{1}{1+\frac{1}{\sigma_{i,S}}} & \text{else} \end{cases}.$$

Namely, we can compute $\sigma_{i,S\cup\{i\}}$ using $\sigma_{i,S}$. Therefore, we have that

$$\text{cost of approximating } \sigma_{i,S\cup\{i\}} = O^*(\text{nnz}(A)) + O^*(T(k)).$$

Using Lemma 6.2.6, we have now

$$T(n) = O^*(\text{nnz}(A) + T(k) + T(\frac{nd}{k}\log d)).$$

Picking $k = \sqrt{nd\log d}$, we have that

$$T(n) = O^*(\text{nnz}(A) + T(\sqrt{nd\log d})).$$

Repeating this process $\tilde{O}(1)$ times, we have

$$T(n) = O^*(\text{nnz}(A) + T(O(d\log d))).$$

## ■ 6.3 Stochastic Gradient Descent

In this section, we study problems of the form $\sum f_i$ where each $f_i$ are convex and the sum can be either infinite or finite.

### Stochastic Problem

In this part, we discuss the problem

$$\min_x F(x) \stackrel{\text{def}}{=} \mathbf{E}_{f\sim\mathcal{D}} f(x)$$

where $\mathcal{D}$ is a distribution of convex functions in $\mathbb{R}^d$. The goal is to find the minimizer $x^* = \text{argmin}_x F(x)$. Suppose we observed $f_1, f_2, \cdots, T$ samples from $\mathcal{D}$. Ideally, we wish to approximate $x^*$ by the empirical risk minimizer

$$x^{(T)}_{\text{ERM}} = \arg\min_x F_T(x) \stackrel{\text{def}}{=} \frac{1}{T}\sum_{i=1}^{T} f_i(x).$$

It is known that $x^{(T)}_{\text{ERM}}$ is optimal in a certain sense albeit its computational cost. Therefore, to discuss the efficiency of an optimization algorithm for $F(x)$, it is helpful to consider the ratio

$$\frac{\mathbf{E}F(x^{(T)}) - F(x^*)}{\mathbf{E}F(x^{(T)}_{\text{ERM}}) - F(x^*)}.$$

We will first discuss the term $\mathbf{E}F(x^{(T)}_{\text{ERM}}) - F(x^*)$. As an example, consider the simplest 1 dimension problem $F(x) = \mathbf{E}_{b\sim N(0,I)}(x-b)^2$. Note that $x^{(T)}$ is simply average of $T$ standard normal variables and hence

$$\begin{aligned}\mathbf{E}F(x^{(T)}_{\text{ERM}}) - F(x^*) &= \mathbf{E}_{b_1,b_2,\cdots,b_T}\mathbf{E}_b(x^{(T)}_{\text{ERM}} - b)^2 - b^2 \\ &= \mathbf{E}_{b_1,b_2,\cdots,b_T}(x^{(T)}_{\text{ERM}})^2 = \frac{1}{T}.\end{aligned}$$

In general, the following lemma shows that

$$\mathbf{E}F(x^{(T)}_{\text{ERM}}) - F(x^*) \to \frac{\sigma^2}{T} \quad \text{where } \sigma^2 = \frac{1}{2}\mathbf{E}\|\nabla f(x^*)\|^2_{(\nabla^2 F(x^*))^{-1}}.$$

where $\frac{\sigma^2}{T}$ is called Cramer-Rao bound.

**Lemma 6.3.1.** *Suppose that $f$ is $\mu$-strongly convex with Lipschitz Hessian for all $f \sim \mathcal{D}$ for some $\mu > 0$. Suppose that $\mathbf{E}_{f\sim\mathcal{D}}\|\nabla f(x^*)\|^2 < +\infty$. Then, we have that*

$$\lim_{N\to+\infty} \frac{\mathbf{E}F(x_{\mathrm{ERM}}^{(N)}) - F(x^*)}{\sigma^2/N} = 1.$$

*Remark.* The statement holds with much weaker assumption. Furthermore, the rate of convergence can be made quantitative.

*Proof.* We first prove $x_{\mathrm{ERM}}^{(N)} \to x^*$ as $N \to +\infty$. By the optimality condition of $x_{\mathrm{ERM}}^{(N)}$, we have that

$$0 = \nabla F_N(x_{\mathrm{ERM}}^{(N)}) = \nabla F_N(x^*) + \nabla^2 F_N(\widetilde{x})(x_{\mathrm{ERM}}^{(N)} - x^*). \tag{6.3.1}$$

By the $\mu$-strongly convexity, we have

$$\|x_{\mathrm{ERM}}^{(N)} - x^*\|_2^2 \le \frac{1}{\mu}\|\nabla F_N(x^*)\|^2.$$

Since $\mathbf{E}_{f\sim\mathcal{D}}\nabla f(x^*) = \nabla F(x^*) = 0$, we have

$$\begin{aligned}\mathbf{E}\|x_{\mathrm{ERM}}^{(N)} - x^*\|_2^2 &\le \frac{1}{\mu}\mathbf{E}\frac{1}{N^2}\sum_i \|\nabla f_i(x^*)\|^2 \\ &= \frac{1}{\mu N}\mathbf{E}_{f\sim\mathcal{D}}\|\nabla f(x^*)\|^2\end{aligned}$$

Therefore, $x_{\mathrm{ERM}}^{(N)} \to x^*$ as $N \to +\infty$.

Now, to compute the error, Taylor expansion of $F$ at $x^*$ shows that

$$F(x_{\mathrm{ERM}}^{(N)}) - F(x^*) = \frac{1}{2}(x_{\mathrm{ERM}}^{(N)} - x^*)^\top \nabla^2 F(\overline{x})(x_{\mathrm{ERM}}^{(N)} - x^*)$$

for some $\overline{x}$ between $x^*$ and $x_{\mathrm{ERM}}^{(N)}$. Using this and (6.3.1) gives

$$F(x_{\mathrm{ERM}}^{(N)}) - F(x^*) = \frac{1}{2}\nabla F_N(x^*)^\top(\nabla^2 F_N(\widetilde{x}))^{-1}\nabla^2 F(\overline{x})(\nabla^2 F_N(\widetilde{x}))^{-1}\nabla F_N(x^*).$$

Since $x_{\mathrm{ERM}}^{(N)} \to x^*$ and $\nabla^2 F_N \succeq \mu I$, we have $(\nabla^2 F_N(\widetilde{x}))^{-1}\nabla^2 F(\overline{x})(\nabla^2 F_N(\widetilde{x}))^{-1} \to (\nabla^2 F(x^*))^{-1}$. Hence, we have

$$\begin{aligned}\lim_{N\to\infty} \mathbf{E}N\cdot(F(x_{\mathrm{ERM}}^{(N)}) - F(x^*)) &= \lim_{N\to\infty}\frac{N}{2}\mathbf{E}\nabla F_N(x^*)^\top(\nabla^2 F(x^*))^{-1}\nabla F_N(x^*) \\ &= \frac{1}{2}\mathbf{E}\|\nabla f(x^*)\|^2_{(\nabla^2 F(x^*))^{-1}}.\end{aligned}$$

□

Now, we discuss how to achieve a bound similar to $\sigma^2/T$ using stochastic gradient descent. Since gradient descent is first order method, we can only achieve a bound related to $\ell_2$ norm $\mathbf{E}\|\nabla f(x^*)\|_2^2$ instead of the inverse Hessian norm. First, we need a lemma relating $\nabla f(x^*)$ and $\nabla f(x)$.

**Exercise 6.3.2.** Suppose $f$ has $L$-Lipschitz gradient. Then, $\|\nabla f(y) - \nabla f(y)\|^2 \le 2L \cdot D_f(y,x)$ for any $x, y$.

**Lemma 6.3.3.** *Suppose $f$ has $L$-Lipschitz gradient for all $f \in \mathcal{D}$. Let $x^*$ be the minimizer of $F$. Then, we have*

$$\mathbf{E}_{f\sim\mathcal{D}}\|\nabla f(x) - \nabla f(x^*)\|_2^2 \le 2L\cdot(F(x) - F(x^*)).$$

*Proof.* By the exercise above, we have

$$\|\nabla f(x) - \nabla f(x^*)\|^2 \le 2L\cdot(f(x) - f(x^*) - \nabla f(x^*)^\top(x - x^*)).$$

Taking expectation, we have the result. □

**Algorithm 20:** `StochasticGradientDescent` (SGD)

```
Input: Initial point $x^{(0)} \in \mathbb{R}^d$, step size $h > 0$.
for $k = 0, 1, \cdots, T$ do
    Sample $f \sim \mathcal{D}$.
    $x^{(k+1)} \leftarrow x^{(k)} - h \cdot \nabla f(x^{(k)})$.
end
```

**Theorem 6.3.4.** *Suppose $f$ has $L$-Lipschitz gradient for all $f \in \mathcal{D}$ and $F$ is $\mu$ strongly convex. Let $x^*$ be the minimizer of $F$ and $\sigma^2 = \frac{1}{2}\mathbf{E}\|\nabla f(x^*)\|^2$. For step size $h \leq \frac{1}{4L}$, the sequence $x^{(k)}$ in* `StochasticGradientDescent` *satisfies*

$$\mathbf{E}\|x^{(k)} - x^*\|^2 \leq 2\frac{h\sigma^2}{\mu} + (1 - \frac{h\mu}{2})^k \cdot \|x^{(0)} - x^*\|^2$$

*and*

$$\frac{1}{T}\sum_{k=0}^{T-1} \mathbf{E}F(x^{(k)}) - F(x^*) \leq h\sigma^2 + \frac{F(x^{(0)}) - F(x^*)}{\mu h T}.$$

*Proof.* Note that

$$\|x^{(k+1)} - x^*\|^2 = \|x^{(k)} - x^*\|^2 - 2h\left\langle x^{(k)} - x^*, \nabla f(x^{(k)})\right\rangle + h^2\|\nabla f(x^{(k)})\|^2.$$

Taking expectation on $f$ and using that

$$\begin{aligned}\mathbf{E}\|\nabla f(x^{(k)})\|^2 &\leq 2\mathbf{E}\|\nabla f(x^{(k)}) - \nabla f(x^*)\|^2 + 2\mathbf{E}\|\nabla f(x^*)\|^2 \\ &\leq \sigma^2 + 4L \cdot (F(x^{(k)}) - F(x^*)),\end{aligned}$$

we have

$$\begin{aligned}\mathbf{E}\|x^{(k+1)} - x^*\|^2 &= \|x^{(k)} - x^*\|^2 - 2h\left\langle x^{(k)} - x^*, \nabla F(x^{(k)})\right\rangle + h^2\sigma^2 + 4Lh^2 \cdot (F(x^{(k)}) - F(x^*)) \\ &\leq \|x^{(k)} - x^*\|^2 - 2\left(h - 2Lh^2\right)(F(x^{(k)}) - F(x^*)) + h^2\sigma^2. \qquad (6.3.2)\end{aligned}$$

Using $h \leq \frac{1}{4L}$ and $F(x^{(k)}) - F(x^*) \leq \frac{1}{2\mu}\|\nabla F(x^{(k)})\|_2^2$, we have

$$\mathbf{E}\|x^{(k+1)} - x^*\|^2 \leq h^2\sigma^2 + (1 - \frac{h\mu}{2}) \cdot \|x^{(k)} - x^*\|^2.$$

The first conclusion follows.

For the second conclusion, (6.3.2) shows that

$$\mathbf{E}F(x^{(k)}) - F(x^*) \leq \frac{\mathbf{E}\|x^{(k)} - x^*\|^2 - \mathbf{E}\|x^{(k+1)} - x^*\|^2}{h} + h\sigma^2.$$

Hence, we have

$$\begin{aligned}\frac{1}{T}\sum_{k=0}^{T-1} \mathbf{E}F(x^{(k)}) - F(x^*) &\leq \frac{\|x^{(0)} - x^*\|^2}{hT} + h\sigma^2 \\ &\leq h\sigma^2 + \frac{F(x^{(0)}) - F(x^*)}{\mu h T}\end{aligned}$$

□

**Exercise 6.3.5.** Applying the theorem above twice with different step size, show one can achieve the distance $\tilde{O}(\frac{\sigma^2}{\mu^2 T})$ and error $\tilde{O}(\frac{L\sigma^2}{\mu^2 T})$, which is $\frac{L}{\mu}$ factor larger than the Cramer-Rao bound.

We note that many algorithms in this book, such as mirror descent, has the stochastic version by replacing the gradient $\nabla F$ by the gradient of the sample $\nabla f$.

## Finite Sum Problem

When $\nabla f(x^*) = 0$ for all $f \sim \mathcal{D}$, stochastic gradient descent algorithm converges exponentially. When the function is given by a finite sum $F = \frac{1}{n}\sum f_i$, we can reduce the variance at $x^*$ by replacing

$$\nabla \widetilde{f}_i(x) = \nabla f_i(x) - \nabla f_i(x^{(0)}) + \nabla F(x^{(0)}).$$

Note that if $x^{(0)}$ is close to $x^*$, then $\nabla \widetilde{f}_i(x^*)$ is small because both $\nabla f_i(x^*) - \nabla f_i(x^{(0)})$ and $\nabla F(x^{(0)})$ are small. Formally, the variance for $\widetilde{f}$ is bounded as follows:

**Lemma 6.3.6.** *Suppose $f$ has L-Lipschitz gradient for all $f \in \mathcal{D}$. For any $f \in \mathcal{D}$, let $\nabla \widetilde{f}(x) = \nabla f(x) - \nabla f(x^{(0)}) + \nabla F(x^{(0)})$ for some fixed $x^{(0)}$. Then, we have*

$$\mathbf{E}\|\nabla \widetilde{f}(x^*)\|^2 \leq 8L \cdot (F(x^{(0)}) - F(x^*)).$$

*Proof.* Note that

$$\begin{aligned}\mathbf{E}\|\nabla \widetilde{f}(x^*)\|^2 &\leq 2\mathbf{E}\|\nabla f(x^*) - \nabla f(x^{(0)})\|^2 + 2\|\nabla F(x^{(0)}) - \nabla F(x^*)\|^2 \\ &= 2\mathbf{E}\|\nabla f(x^*) - \nabla f(x^{(0)})\|^2 + 2\mathbf{E}\|\nabla f(x^{(0)}) - \nabla f(x^*)\|^2 \\ &\leq 4L(F(x^{(0)}) - F(x^*)) + 4L(F(x^{(0)}) - F(x^*))\end{aligned}$$

where we used Lemma 6.3.3 at the end. □

**Algorithm 21:** `StochasticVarianceReducedGradient` (SVRG)

```
Input: Initial point $x^{(0)} \in \mathbb{R}^d$, step size $h > 0$, restart time $m$.
$\widetilde{x} = x^{(0)}$.
for $k = 0, 1, \cdots, T$ do
    if $k$ is a multiple of $m$ and $k \neq 0$ then
        $x^{(k)} = \frac{1}{m}\sum_{j=1}^{m} x^{(k-j)}$.
        $\widetilde{x} = x^{(k)}$.
    end
    Sample $f_i$ for $i \in [n]$
    $x^{(k+1)} \leftarrow x^{(k)} - h \cdot (\nabla f(x^{(k)}) - \nabla f(\widetilde{x}) + \nabla F(\widetilde{x}))$.
end
```

**Theorem 6.3.7.** *Suppose $f$ has L-Lipschitz gradient for all $f_i$ and $F$ is $\mu$ strongly convex. Let $x^*$ be the minimizer of $F$. For step size $h = \frac{1}{32L}$ and $m = \frac{128L}{\mu}$, the sequence $x^{(km)}$ in* `StochasticVarianceReducedGradient` *satisfies*

$$\mathbf{E}F(x^{(km)}) - F(x^*) \leq \frac{1}{2^k} \cdot (F(x^{(0)}) - F(x^*)).$$

*In particular, it takes $O((n + \frac{L}{\mu})\log(\frac{1}{\epsilon}))$ gradient computation to find $x$ such that $\mathbf{E}F(x) - F(x^*) \leq \epsilon \cdot (F(x^{(0)}) - F(x^*))$.*

*Remark.* Gradient descent gives $O(n\frac{L}{\mu}\log(\frac{1}{\epsilon}))$ instead because computing $\nabla F$ involves computing $n$ many $\nabla f$.

*Proof.* The algorithm consists of $\frac{T}{m}$ phases. For the first phase, we compute $\nabla F(x^{(0)})$. Lemma 6.3.6 shows that

$$\mathbf{E}\|\nabla f(x^{(k)}) - \nabla f(x^{(0)}) + \nabla F(x^{(0)})\|^2 \leq 8L \cdot (F(x^{(0)}) - F(x^*)).$$

Hence, Theorem 6.3.4 shows that

$$\begin{aligned}\frac{1}{m}\sum_{k=0}^{m-1} \mathbf{E}F(x^{(k)}) - F(x^*) &\leq 8hL \cdot (F(x^{(0)}) - F(x^*)) + \frac{F(x^{(0)}) - F(x^*)}{\mu h m} \\ &\leq \frac{1}{2}(F(x^{(0)}) - F(x^*)).\end{aligned}$$

Hence, the error decreases by half each phase. This shows the result. □

## 6.4 Coordinate Descent

If some of the coordinates are more important than other, it makes sense to update the important coordinates more often than others.

**Lemma 6.4.1.** *Given a convex function $f$. Suppose that $\frac{\partial^2}{\partial x_i^2} f(x) \leq L_i$ for all $x$ and let $L = \sum L_i$. If we sample coordinate $i$ with probability $p_i = \frac{L_i}{L}$, then,*

$$\mathbf{E}_i f(x - \frac{1}{L_i}\frac{\partial}{\partial x_i} f(x) e_i) \leq f(x) - \frac{1}{2L}\|\nabla f(x)\|_2^2 .$$

*Proof.* Note that the function $\zeta(t) = f(x + te_i)$ is $L_i$ smooth. Hence, we have that

$$f(x - \frac{1}{L_i}\frac{\partial}{\partial x_i} f(x) e_i) \leq f(x) - \frac{1}{2L_i}\frac{\partial}{\partial x_i} f(x)^2.$$

Since we sample coordinate $i$ with probability $p_i = \frac{L_i}{L}$, we have that

$$\begin{aligned}\mathbf{E} f(x - \frac{1}{L_i}\frac{\partial}{\partial x_i} f(x) e_i) &= f(x) - \sum_i \frac{L_i}{L}\frac{1}{2L_i}\frac{\partial}{\partial x_i} f(x)^2 \\ &= f(x) - \frac{1}{2L}\sum_i \frac{\partial}{\partial x_i} f(x)^2 \\ &= f(x) - \frac{1}{2L}\|\nabla f(x)\|_2^2 .\end{aligned}$$

□

By the same proof as Theorem 1.5.5, we have the following:

**Algorithm 22:** `CoordinateDescent` (CD)

```
Input: Initial point $x^{(0)} \in \mathbb{R}^d$, step size $h > 0$.
for $k = 0, 1, \cdots, T$ do
    Sample $i$ with probability proportional to $L_i$.
    $x^{(k+1)} \leftarrow x^{(k)} - \frac{1}{L_i}\frac{\partial}{\partial x_i} f(x^{(k)}) e_i$.
end
```

**Theorem 6.4.2** (Coordinate Descent Convergence)**.** *Given a convex function $f$. Suppose that $\frac{\partial^2}{\partial x_i^2} f(x) \leq L_i$ for all $x$ and let $L = \sum L_i$. Consider the algorithm $x^{(k+1)} \leftarrow x^{(k)} - \frac{1}{L_i}\frac{\partial}{\partial x_i} f(x^{(k)}) e_i$, we have that*

$$\mathbf{E} f(x^{(k)}) - f(x^*) \leq \frac{2LR^2}{k+4} \quad \textit{with} \quad R = \max_{f(x) \leq f(x^{(0)})} \|x - x^*\|_2$$

*for any minimizer $x^*$ of $f$. If $f$ is $\mu$ strongly convex, then we also have*

$$\mathbf{E} f(x^{(k)}) - f(x^*) \leq (1 - \frac{\mu}{L})^{\Omega(k)}(f(x^{(0)}) - f(x^*)).$$

*Remark.* Note that $\frac{L}{d} \preceq \nabla^2 f(x) \preceq L \cdot I$. Therefore gradient descent takes at least $\frac{1}{d}$ many steps while each step takes $d$ times longer (usually).

### Application to Laplacian systems (in progress)

Given a graph $G = (V, E)$ with $m$ edges and $n$ vertices. We consider the problem

$$\frac{1}{2}\sum_{(i,j)\sim E}(x_i - x_j)^2 - c^\top x.$$

For now, we can think this as a generalization of the worst case function. Directly applying accelerated gradient descent, accelerated coordinate descent or accelerated stochastic descent all gives $O^*(mn)$ time algorithm. Roughly speaking, this is because of the "diameter" bottleneck for all first-order methods.

To get around the diameter issue, we let the incidence matrix $B \in \mathbb{R}^{|E|\times|V|}$ defined by $B_{(i,j),i} = 1$ and $B_{(i,j),j} = -1$.We consider the dual problem

$$\min_{B^T f = d} \frac{1}{2} \sum_{e\in E} f_e^2.$$

Let pick a tree $T$, by sending flow along the tree, it is easy to find $g$ such that $B^\top g = d$. Hence, the problem can be rewritten as

$$\min_{B^T f = 0} \frac{1}{2} \sum_{e\in E} (f+g)_e^2.$$

We call any $f$ satisfying $B^\top f = 0$ is a circulation. For any edge $e \notin T$, we define $c(e)$ be an unit cycle on $\{e\} \cup T$. It is known that any circulation can be represented by

$$f = \sum_{e\notin T} \alpha_e c(e).$$

Also, any such flow is a circulation. Therefore, the problem now becomes

$$\min_{\alpha} \ell(\alpha) \stackrel{\text{def}}{=} \frac{1}{2} \sum_{e\in E} (\sum_{k\notin T} \alpha_k c(k) + g_e)_e^2.$$

Now, we apply accelerated coordinate descent to solve this problem. First, we note that $\ell$ is 1 strongly convex. Next, for any $k$, $\frac{\partial^2}{\partial \alpha_k^2} \ell = \text{length}(c(k))$. Hence, the accelerated coordinate descent takes $O^*(\sum_{k\notin T} \text{length}(c(k)))$ coordinate steps. It is known that for any graph $G$, we can find a tree such that $\sum_{k\notin T} \text{length}(c(k)) = O^*(m)$. Using such tree, the running time is $O^*(m)$ coordinate steps. It is also easy to implement the algorithm such that each step takes $O^*(1)$ time. Therefore, this gives a $O^*(m)$ algorithm for the problem. See [32] for more details.

Chapter 7

# Multiplication

## ■ 7.1 Simulated Annealing

In this chapter we study a sampling-based approach for optimization and integration in high dimension. The main idea is to sample a sequence of logconcave distributions, starting with one that is easy to integrate and ending with the function whose integral is desired. This process is known as *simulated annealing.* The same high-level algorithm can be used for optimization, volume computation/integration or rounding. For integration, the desired integral can be expressed as the following telescoping product:

$$\int_{\mathbb{R}^n} f = \int f_0 \frac{\int f_i}{\int f_0} \frac{\int f_2}{\int f_3} \cdots \frac{\int f_m}{\int f_{m-1}}$$

where $f_m = f$. Each ratio $\int f_{i+1} / \int f_i$ is the expectation of the estimator $Y = \frac{f_{i+1}(X)}{f_i(X)}$ for $X$ drawn from the density proportional to $f_i$. What sequence of functions should we use?

```
Algorithm 23: SimulatedAnnealing
  1. For $i = 0, \ldots, m$, define
         $a_i = b\left(1 + \frac{1}{\sqrt{n}}\right)^i$ and $f_i(x) = f(x)^{a_i}$.
  2. Let $X_0^1, \ldots, X_0^k$ be independent random points with density proportional to $f_0$.
  3. For $i = 0, \ldots, m-1$: starting with $X_i^1, \ldots, X_i^k$, generate random points $X_{i+1} = \{X_{i+1}^1, \ldots, X_{i+1}^k\}$; update a
     running estimate $g$ based on these samples; update the isotropy transformation using the samples.
  4. Output the final estimate of $g$.
return $x$.
```

For optimization, the function $f_m$ is set to be a sufficiently high power of $f$, the function to be maximized while $g$ is simply the maximum objective value so far. For integration and rounding, $f_m = f$, the target function to be integrated or rounded. For integration, the function $g$ starts out as the integral of $f_0$ and is multiplied by the ratio $\int f_{i+1} / \int f_i$ in each step. For rounding, $g$ is simply the estimate of the isotropic transformation for the current function.

For sampling and optimization, it is natural to ask why one uses a sequence of functions rather than jumping straight to the target function $f_m$. For optimizing a linear function $c^T x$ over a convex set $K$, all we need to do is sample according to the density proportional to $e^{-a(c^T x)}$ for a sufficiently large coefficient $a$ (recall Theorem 3.5.1). The reason for using a sequence is that the function $f_m$ and corresponding density $p_m$ can be very far from the starting function or distribution, and hence the mixing time can be high. The sequence ensures that samples from the current function provide a good (warm) start for sampling from the next function. This is captured in the next lemma.

**Theorem 7.1.1.** *Let $p_i(x) = f_i(x) / \int f_i$ with $f_i$ defined as in the annealing algorithm for some logconcave function*

*$f : \mathbb{R}^n \to \mathbb{R}_+$. Then a random sample $X \sim p_i$ satisfies*

$$\mathbf{E}_{p_i}\left(\frac{p_i(x)}{p_{i+1}(x)}\right) = O(1).$$

The underlying mathematical property behind this lemma is the following.

**Lemma 7.1.2.** *Let $f$ be a logconcave function in $\mathbb{R}^n$. Then $Z(a) = a^n \int_{\mathbb{R}^n} f(x)^a$ is logconcave for $a \geq 0$. If $f$ has support $K$, then $Z(a) = a^n \int_K f(ax)$ is logconcave for $a > 0$.*

## 7.2 Volume Computation

For volume computation, we can apply annealing as follows. We assume that the input convex body $K$ contains a unit ball, has diameter bounded by $D$ and is given by a membership oracle. The polynomial-time algorithm of Dyer, Frieze and Kannan [18] uses a sequence of uniform distributions on convex bodies, starting with the ball contained inside the input body $K$. Each body in the sequence is a ball intersected with the given convex body $K$: $K_i = 2^{\frac{i}{n}} rB \cap K$ and $f_i(x)$ is the indicator of $K_i$. The length the sequence is $m = O(n \log D)$ so that the final body is just $K$. A variance computation shows that $O(m/\epsilon^2)$ samples per distribution suffice to get an overall $1+\epsilon$ multiplicative error approximation with high probability. The total number of samples is $O^*(m^2) = O^*(n^2)$ and the complexity of the resulting algorithm is $O^*(n^5)$ as shown in [25]. Table 7.1 below summarizes progress on the volume problem over the past three decades. Besides improving the complexity of volume computation, each step has typically resulted in new techniques. For more details, we refer the reader to surveys on the topic [57, 61].

| Year/Authors | New ingredients | Steps |
|---|---|---|
| 1989/Dyer-Frieze-Kannan [18] | Everything | $n^{23}$ |
| 1990/Lovász-Simonovits [39] | Better isoperimetry | $n^{16}$ |
| 1990/Lovász [38] | Ball walk | $n^{10}$ |
| 1991/Applegate-Kannan [4] | Logconcave sampling | $n^{10}$ |
| 1990/Dyer-Frieze [17] | Better error analysis | $n^8$ |
| 1993/Lovász-Simonovits [40] | Localization lemma | $n^7$ |
| 1997/Kannan-Lovász-Simonovits [25] | Speedy walk, isotropy | $n^5$ |
| 2003/Lovász-Vempala [42] | Annealing, hit-and-run | $n^4$ |
| 2015/Cousins-Vempala [15] (well-rounded) | Gaussian Cooling | $n^3$ |
| 2017/Lee-Vempala (polytopes) | Hamiltonian Walk | $mn^{\frac{2}{3}}$ |

**Table 7.1:** The complexity of volume estimation, each step uses $\widetilde{O}(n)$ bit of randomness. The last algorithm needs $\widetilde{O}\left(mn^{\omega-1}\right)$ steps per iteration while the rest need $O(n^2)$ per oracle query.

In [42] this was improved by sampling from a sequence of nonuniform distributions. Then we consider the following estimator:

$$Y = \frac{f_{i+1}(X)}{f_i(X)}.$$

We see that

$$\mathbf{E}_{f_i}(Y) = \frac{\int f_{i+1}}{\int f_i}.$$

In the algorithm of DFK and KLS, this ratio is bounded by a constant in each phase, giving a total of $O^*(n)$ phases since the ratio of final to initial integrals is exponential. Instead of uniform densities, we consider

$$f_i(x) \propto \exp(-a_i \|x\|)\chi_K(x) \text{ or } f_i(x) \propto \exp(-a_i \|x\|^2)\chi_K(x).$$

The coefficient $a_i$ (inverse "temperature") will be changed by a factor of $(1+\frac{1}{\sqrt{n}})$ in each phase, which implies that $m = \widetilde{O}(\sqrt{n})$ phases suffice to reach the target distribution. This is perhaps surprising since the ratio of the initial integral to the final is typically $n^{\Omega(n)}$. Yet the algorithm uses only $\widetilde{O}(\sqrt{n})$ phases, and hence estimates a ratio of

$n^{\tilde{\Omega}(\sqrt{n})}$ in one or more phases. The key insight is that even though the expected ratio might be large, its variance is not.

**Lemma 7.2.1.** *For $X \sim f_i$ with $f_i(x) = e^{-a_i\|x\|}\chi_K(x)$ for a convex body $K$, or $f_i(x) = f(x)^{a_i}$ for a logconcave function $f$, we have that the estimator $Y = \frac{f_{i+1}(X)}{f_i(X)}$ satisfies*

$$\frac{\mathbf{E}\left(Y^2\right)}{\mathbf{E}\left(Y\right)} \le \left(\frac{a_{i+1}^2}{(2a_{i+1} - a_i)a_i}\right)^n$$

*which is bounded by a constant for $a_i = a_{i+1}\left(1 + \frac{1}{\sqrt{n}}\right)$.*

**Theorem 7.2.2** ([42])**.** *The volume of a convex body in $\mathbb{R}^n$ (given by a membership oracle) can be computed to relative error $\varepsilon$ using $\widetilde{O}(n^4/\varepsilon^2)$ oracle queries and $\widetilde{O}(n^2)$ arithmetic operations per query.*

The LV algorithm has two parts. In the first it finds a transformation that puts the body in near-isotropic position. The complexity of this part is $\widetilde{O}(n^4)$. In the second part, it runs the annealing schedule, while maintaining that the distribution being sampled is *well-rounded*, a weaker condition than isotropy. Well-roundedness requires that a level set of measure $\frac{1}{8}$ contains a constant-radius ball and the trace of the covariance (expected squared distance of a random point from the mean) to be bounded by $O(n)$, so that $R/r$ is effectively $O(\sqrt{n})$. To achieve the complexity guarantee for the second phase, it suffices to use the KLS bound of $\psi_p \gtrsim n^{-\frac{1}{2}}$. Connecting improvements in the Cheeger constant directly to the complexity of volume computation is an open question. To apply improvements in the Cheeger constant, one would need to replace well-roundedness with (near-)isotropy and maintain that. However, maintaining isotropy appears to be much harder — possibly requiring a sequence of $\Omega(n)$ distributions and $\Omega(n)$ samples from each, providing no gain over the current complexity of $O^*(n^4)$ even if the KLS conjecture turns out to be true.

A faster algorithm is known for well-rounded convex bodies (any isotropic logconcave density satisfies $\frac{R}{r} = O(\sqrt{n})$ and is well-rounded). This variant of simulated annealing, called Gaussian cooling utilizes the fact that the KLS conjecture holds for a Gaussian density restricted by any convex body, and completely avoids computing an isotropic transformation.

**Theorem 7.2.3** ([15])**.** *The volume of a well-rounded convex body, i.e., with $R/r = O^*(\sqrt{n})$, can be computed using $O^*(n^3)$ oracle calls.*

## ■ 7.3 Rounding

**Theorem 7.3.1.** *Any convex body can be put in 2-isotropic position in $O^*(n^4)$ membership oracle queries and $O(n^2)$ additional computation per query.*

Chapter 8

# Acceleration (in progress)

## 8.1 Chebyshev Polynomials

The goal of this section is to introduce some basic results in approximation theory. We will bound $\inf_{q(0)=1} \left(\max_i q(\lambda_i)^2\right)$ for matrices $A$ satisfying $\mu \cdot I \preceq A \preceq L \cdot I$. First, we note that we can choose $q(x) = \left(1 - \frac{x}{L}\right)^k$ and get

$$\inf_{q(0)=1} \max_{\mu \leq x \leq L} q(x)^2 \leq \left(1 - \frac{\mu}{L}\right)^k .$$

This shows that it takes $O(\frac{L}{\mu} \log(\frac{1}{\epsilon}))$ degree to make the error bounded by $\epsilon$. The key fact we will use is that we can in general approximate a polynomial of degree $k$ by a polynomial of degree $\tilde{O}(\sqrt{k})$. The construction involves Chebyshev polynomials. This will imply a $\tilde{O}(\sqrt{\frac{L}{\mu}})$ iteration bound for the conjugate gradient method of the next section.

**Definition 8.1.1.** For any integer $d$, we define the Chebyshev polynomial

$$T_d(x) = \cos(d \cos^{-1} x).$$

**Exercise 8.1.2.** Show that $T_d(x)$ is a degree $|d|$ polynomial in $x$ and that

$$xT_d(x) = \frac{T_{d+1}(x) + T_{d-1}(x)}{2}. \tag{8.1.1}$$

**Theorem 8.1.3** (Chernoff Bound)**.** *For independent random variables $Y_1, \cdots, Y_s$ such that $\mathbf{P}(Y_i = 1) = \mathbf{P}(Y_i = -1) = \frac{1}{2}$, for any $a \geq 0$, we have*

$$\mathbf{P}(\left|\sum_{i=1}^{s} Y_i\right| \geq a) \leq 2\exp(-\frac{a^2}{2s}).$$

Now, we are ready to show that there is a degree $\tilde{O}(\sqrt{s})$ polynomial that estimates $x^s$.

**Theorem 8.1.4.** *For any positive integers $s$ and $d$, there is a polynomial $p$ of degree $d$ such that*

$$\max_{x\in[-1,1]} |p(x) - x^s| \leq 2\exp(-\frac{d^2}{2s}).$$

*Proof.* Let $Y_i$ are i.i.d. random variable uniform on $\{-1, 1\}$. Let $Z_s = \sum_{i=1}^{s} Y_i$. Note that

$$\mathbf{E}_{z\sim Z_s} T_z = \mathbf{E}_{z\sim Z_{s-1}} \frac{1}{2}(T_{z+1} + T_{z-1})$$

Now, (8.1.1) shows that $\mathbf{E}_{z\sim Z_s} T_z(x) = \mathbf{E}_{z\sim Z_{s-1}} xT_z(x)$. By induction, we have

$$\mathbf{E}_{z\sim Z_s} T_z(x) = x^s T_0(x) = x^s. \tag{8.1.2}$$

Now, we define the polynomial $p(x) = \mathbf{E}_{z\sim Z_s} T_z(x) 1_{|z|\leq d}$. The error of the polynomial can be bounded as follows

$$\begin{aligned}\max_{x\in[-1,1]} |p(x) - x^s| &= \max_{x\in[-1,1]} |\mathbf{E}_{z\sim Z_s} T_z(x) 1_{|z|>d}| \\ &\leq \max_{x\in[-1,1]} \mathbf{E}_{z\sim Z_s} |T_z(x)| 1_{|z|>d} \\ &\leq \mathbf{P}_{z\sim Z_s}(|z| > d) \\ &\leq 2\exp(-\frac{d^2}{2s})\end{aligned}$$

where we used (8.1.2) on the first line, $|T_z(x)| \leq 1$ for all $x \in [-1, 1]$ on the third line and Chernoff bound (Theorem 8.1.3) on the last line. $\square$

*Remark.* This proof comes from [53]. Please see [53] for the approximation of other functions.

By scaling and translating Theorem 8.1.4, we have the following:

**Theorem 8.1.5.** *For any $0 < \mu \leq L$ and any $0 \leq \epsilon \leq 1$, there is a polynomial $q$ of degree $O(\sqrt{\frac{L}{\mu}} \log(\frac{1}{\epsilon}))$ such that*

$$\inf_{q(0)=1} \max_{\mu\leq x\leq L} q(x)^2 \leq \epsilon.$$

*Proof.* Theorem 8.1.4 shows that there is $q$ such that

$$\max_{x\in[0,L]} \left| q(x) - \left(1 - \frac{x}{L}\right)^s \right| \leq 2\exp(-\frac{d^2}{2s}).$$

Hence, we have that $|q(0) - 1| \leq 2\exp(-\frac{d^2}{2s})$ and that

$$\max_{x\in[\mu,L]} |q(x)| \leq \left(1 - \frac{\mu}{L}\right)^s + 2\exp(-\frac{d^2}{2s}).$$

To make both $|q(0) - 1| \leq \frac{\epsilon}{3}$ and $\max_{x\in[\mu,L]} |q(x)| \leq \frac{\epsilon}{3}$, we set $d = O(\sqrt{s \log \frac{1}{\epsilon}})$ and $s = O(\frac{L}{\mu} \log(\frac{1}{\epsilon}))$. By rescaling $q$, we have $q(0) = 1$. $\square$

## 8.2 Conjugate Gradient

Consider any algorithm starts with $x^{(0)} = 0$ and satisfies the invariant

$$x^{(k+1)} \in x^{(0)} + \mathrm{span}\{\nabla f(x^{(0)}), \nabla f(x^{(1)}), \cdots, \nabla f(x^{(k)})\}.$$

Then, it is easy to see that $x^{(k)} \in \mathcal{K}_k$ defined as follows:

**Definition 8.2.1.** Define the Krylov subspaces $\mathcal{K}_0 = \{0\}$, $\mathcal{K}_k = \mathrm{span}\{b, Ab, \cdots, A^{k-1}b\}$.

Therefore, the best such an algorithm can do is $\min_{x\in\mathcal{K}_k} f(x)$. For quadratic functions $f(x) \overset{\text{def}}{=} \frac{1}{2}x^\top Ax - b^\top x$, one can compute the $\min_{x\in\mathcal{K}_k} f(x)$ efficiently using conjugate gradient. Without loss of generality, we can assume $A$ is a symmetric $n \times n$ matrix.

**Lemma 8.2.2.** *Let $x^{(k)} = \mathrm{argmin}_{x\in\mathcal{K}_k} f(x)$ be the Krylov sequence. . Then, the steps $v^{(k)} = x^{(k)} - x^{(k-1)}$ are "conjugate", namely,*

$$v^{(i)\top} A v^{(j)} = 0 \text{ for all } i \neq j.$$

*Proof.* Assume that $i < j$. The optimality of $x^{(j)}$ shows that $\nabla f(x^{(j)}) \in \mathcal{K}_j^\perp \subset \mathcal{K}_{j-1}^\perp$ and that $\nabla f(x^{(j-1)}) \in \mathcal{K}_{j-1}^\perp$. Hence, we have

$$Av^{(j)} = \nabla f(x^{(j)}) - \nabla f(x^{(j-1)}) \in \mathcal{K}_{j-1}^\perp.$$

Next, we note that $v^{(i)} = x^{(i)} - x^{(i-1)} \in \mathcal{K}_i \subset \mathcal{K}_{j-1}$. Hence, we have $v^{(i)\top} A v^{(j)} = 0$. $\square$

Since the steps are conjugate, $v^{(i)}$ forms a conjugate basis for the Krylov subspaces:

$$\mathcal{K}_k = \text{span}\{v^{(1)}, v^{(2)}, \cdots, v^{(k)}\}.$$

Note that $x^{(k)} = x^{(k-1)} + v^{(k)}$. Hence, it suffices to find a formula for $v^{(k)}$.

**Lemma 8.2.3.** *We have that*

$$v^{(k)} = \frac{v^{(k)\top} r^{(k-1)}}{\|r^{(k-1)}\|^2} \left( r^{(k-1)} - \frac{r^{(k-1)\top} A v^{(k-1)}}{v^{(k-1)} A v^{(k-1)}} v^{(k-1)} \right).$$

*Proof.* Again, the optimality condition shows that $\nabla f(x^{(k-1)}) \in \mathcal{K}_{k-1}^{\perp}$ and that $\nabla f(x^{(k-1)}) \in \mathcal{K}_k$ by the definition of $\mathcal{K}$. Hence, we have

$$\mathcal{K}_k = \text{span}\{v^{(1)}, \cdots, v^{(k-1)}, r^{(k-1)}\}$$

where $r^{(k-1)} = b - Ax^{(k-1)}$. Therefore, we have

$$v^{(k)} = c_0 r^{(k-1)} + \sum_{i=1}^{k-1} c_i v^{(i)}$$

for some $c_0$.

For $c_0$, we use that $r^{(k-1)} \in \mathcal{K}_{k-1}^{\perp}$. This gives $v^{(k)\top} r^{(k-1)} = c_0 \|r^{(k-1)}\|^2$.

For $c_{k-1}$, since $v^{(i)\top} A v^{(k-1)} = 0$ for any $i \neq k-1$, we have

$$0 = v^{(k)\top} A v^{(k-1)} = c_0 r^{(k-1)\top} A v^{(k-1)} + c_{k-1} v^{(k-1)} A v^{(k-1)}$$

and $c_{k-1} = -c_0 \cdot \frac{r^{(k-1)\top} A v^{(k-1)}}{v^{(k-1)} A v^{(k-1)}}$.

For other $c_i$, we note that $Av^{(j)} \in \mathcal{K}_{j+1}$ and $r^{(k-1)} \in \mathcal{K}_{k-1}^{\perp}$. Hence, $c_i = 0$ for all $j \notin \{0, k-1\}$.

Substitute the $c_i$, we get

$$v^{(k)} = \frac{v^{(k)\top} r^{(k-1)}}{\|r^{(k-1)}\|^2} \left( r^{(k-1)} - \frac{r^{(k-1)\top} A v^{(k-1)}}{v^{(k-1)} A v^{(k-1)}} v^{(k-1)} \right).$$

□

To make the formula simpler, we define $p^{(k)} = \frac{\|r^{(k-1)}\|_2^2}{v^{(k)\top} r^{(k-1)}} v^{(k)}$.

**Lemma 8.2.4.** *We have that* $x^{(k)} = x^{(k-1)} + \frac{\|r^{(k-1)}\|_2^2}{p^{(k)\top} A p^{(k)}} p^{(k)}$ *and* $p^{(k)} = r^{(k-1)} - \frac{\|r^{(k-1)}\|^2}{\|r^{(k-2)}\|_2^2} p^{(k-1)}$.

*Proof.* For the first formula, note that

$$x^{(k)} = x^{(k-1)} + v^{(k)} = x^{(k-1)} + \frac{v^{(k)\top} r^{(k-1)}}{\|r^{(k-1)}\|_2^2} p^{(k)}.$$

For the quantity $v^{(k)\top} r^{(k-1)}$, we note that $f(x^{(k-1)} + t v^{(k)})$ is minimized at $t = 1$ and hence

$$\begin{aligned} v^{(k)\top} r^{(k-1)} &= v^{(k)\top} A v^{(k)} \\ &= \frac{v^{(k)\top} r^{(k-1)}}{\|r^{(k-1)}\|_2^2} \frac{v^{(k)\top} r^{(k-1)}}{\|r^{(k-1)}\|_2^2} p^{(k)\top} A p^{(k)} \end{aligned}$$

where we used the definition of $p^{(k)}$. Simplifying gives

$$\frac{v^{(k)\top} r^{(k-1)}}{\|r^{(k-1)}\|_2^2} = \frac{\|r^{(k-1)}\|_2^2}{p^{(k)\top} A p^{(k)}}$$

and hence the first formula.

For the second formula, we use Lemma 8.2.3 and substitute $p^{(k)} = \frac{\|r^{(k-1)}\|_2^2}{v^{(k)\top}r^{(k-1)}}v^{(k)}$. This gives

$$p^{(k)} = r^{(k-1)} - \frac{r^{(k-1)\top}Av^{(k-1)}}{\|r^{(k-2)}\|_2^2}p^{(k-1)}. \tag{8.2.1}$$

Note that

$$r^{(k-1)} = b - Ax^{(k-1)} = r^{(k-2)} - Av^{(k-1)}.$$

Taking inner product with $r^{(k-1)}$ and using $r^{(k-1)} \perp r^{(k-2)}$ gives $\|r^{(k-1)}\|^2 = r^{(k-1)\top}Av^{(k-1)}$. Put this into (8.2.1) gives the result. □

**Algorithm 24:** `ConjugateGradient` (CG)

```
$r^{(0)} = b - Ax^{(0)}$. $p^{(0)} = r^{(0)}$.
for $k = 1, 2, \cdots$ do
    $\alpha = \frac{\|r^{(k-1)}\|_2^2}{p^{(k)\top}Ap^{(k)}}$.
    $x^{(k)} \leftarrow x^{(k-1)} + \alpha p^{(k)}$.
    $r^{(k)} \leftarrow r^{(k-1)} + \alpha Ap^{(k)}$.
    if $\|r^{(k)}\|_2 \le \epsilon$ then return $x^{(k)}$
    $p^{(k+1)} = r^{(k)} - \frac{\|r^{(k)}\|^2}{\|r^{(k-1)}\|^2}p^{(k)}$.
end
```

**Theorem 8.2.5.** *Let $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ for some symmetric matrix $A$ and some vector $b$. Let $x^{(k)}$ be the sequence produced by* `ConjugateGradient`*. Then, we have*

$$f(x^{(k)}) - f(x^*) \le \frac{1}{2}\inf_{q(0)=1}\max_i q(\lambda_i)^2 \cdot \|b\|_{A^{-1}}^2$$

*where $q$ searches over polynomials of degree at most $k$ and $\lambda_i$ are eigenvalues of $A$.*

*Proof.* Note that $\mathcal{K}_k = \{p(A) : \deg p < k\}$. Hence, we have

$$\begin{aligned}
2(f(x^{(k)}) - f(x^*)) &= \min_{x\in\mathcal{K}_k}\|x - x^*\|_A^2 \\
&= \inf_{\deg p<k}\|(p(A) - A^{-1})b\|_A^2 \\
&= \inf_{\deg p<k}\|A^{-\frac{1}{2}}(Ap(A) - I)A^{-\frac{1}{2}}b\|_A^2 \\
&= \inf_{\deg p<k}\|(Ap(A) - I)A^{-\frac{1}{2}}b\|_2^2
\end{aligned}$$

Let $q(x) = 1 - xp(x)$. Note that $q(0) = 1$ and $\deg q \le k$ and any such $q$ is of the form $1 - xp$. Hence, we have

$$\begin{aligned}
2(f(x^{(k)}) - f(x^*)) &= \inf_{\deg q\le k, q(0)=1}\|q(A)A^{-\frac{1}{2}}b\|_2^2 \\
&\le \inf_{q(0)=1}\|q(A)\|_{\mathrm{op}}^2 \cdot \|b\|_{A^{-1}}^2.
\end{aligned}$$

The result follows from the fact that $\|q(A)\|_{\mathrm{op}}^2 = \max_i q(\lambda_i)^2$. □

It is known that for any $0 < \mu \le L$, there is a degree $k$ polynomial $q$ with $q(0) = 1$ such that

$$\max_i q(\lambda_i)^2 \le 2\left(1 - \frac{2}{\sqrt{\frac{L}{\mu}} + 1}\right)^k$$

Therefore, it takes $O(\sqrt{\frac{L}{\mu}}\log(\frac{1}{\epsilon}))$ iterations to find $x$ such that $f(x) - f(x^*) \le \epsilon \cdot \|b\|_{A^{-1}}^2$.

Also, we note that if there are only $s$ many distinct eigenvalues in $A$, then conjugate gradient finds the exact solution in $s$ iterations.

## 8.3 Accelerated Gradient Descent via Plane Search

According to legend, the first proof for accelerated gradient descent is due to Nemirovski in the 70s. The first proof involves a 2-dimensional plane search subroutine. Later on this was improved to line search and finally Nesterov showed how to get rid all line search in 1982. We change the proof of Nemirovski slightly to get rid of all uses of parameters.

**Algorithm 25:** `NemAGD`

```
Input: Initial point $x^{(1)} \in \mathbb{R}^n$.
for $k = 1, 2, \cdots$ do
    $x^{(k)+} \leftarrow \text{argmin}_{x=x^{(k)}+t\nabla f(x^{(k)})} f(x)$.
    $P^{(k)} \leftarrow x^{(1)} + \text{span}(x^{(k)+} - x^{(1)}, \sum_{s=1}^{k} \frac{\nabla f(x^{(s)})}{\|\nabla f(x^{(s)})\|})$.
    // Alternatively, one can use $P^{(k)} \leftarrow x^{(k)} + \text{span}(x^{(k)} - x^{(1)}, \nabla f(x^{(k)}), \sum_{s=1}^{k} \frac{\nabla f(x^{(s)})}{\|\nabla f(x^{(s)})\|})$ without
       defining $x^{(k)+}$.
    $x^{(k+1)} = \arg\min_{x \in P^{(k)}} f(x)$.
end
```

**Theorem 8.3.1.** *Assume that $f$ is convex with $\nabla^2 f(x) \preceq L \cdot I$ for all $x$. Then, we have that*

$$f(x^{(k)}) - f(x^*) \lesssim \frac{L\|x^* - x^{(1)}\|^2}{k^2}.$$

*Proof.* Let $\delta_k = f(x^{(k)}) - f(x^*)$. By the convexity of $f$, we have

$$\begin{aligned}\delta_k &\leq \nabla f(x^{(k)})^\top (x^{(k)} - x^*) \\ &= \nabla f(x^{(k)})^\top (x^{(k)} - x^{(1)}) + \nabla f(x^{(k)})^\top (x^{(1)} - x^*).\end{aligned}$$

Since $x^{(k)}$ is the minimizer on $P^{(k-1)}$ which contains $x^{(1)}$, we have $\nabla f(x^{(k)})^\top (x^{(k)} - x^{(1)}) = 0$ and hence

$$\delta_k \leq \nabla f(x^{(k)})^\top (x^{(1)} - x^*).$$

Let $\lambda_t = \frac{1}{\|\nabla f(x^{(t)})\|}$. Note that

$$\sum_{k=1}^{T} \lambda_k \delta_k \leq \left\langle \sum_{k=1}^{T} \frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|}, x^{(1)} - x^* \right\rangle \leq \|x^* - x^{(1)}\|_2 \cdot \left\| \sum_{k=1}^{T} \frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|} \right\|_2 .$$

Finally, we note that $\nabla f(x^{(k+1)}) \perp P^{(k)}$ and hence $\nabla f(x^{(k+1)}) \perp \sum_{s=1}^{k} \frac{\nabla f(x^{(s)})}{\|\nabla f(x^{(s)})\|}$. Therefore, we have

$$\left\| \sum_{k=1}^{T} \frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|} \right\|_2^2 = \sum_{k=1}^{T} \left\| \frac{\nabla f(x^{(k)})}{\|\nabla f(x^{(k)})\|} \right\|_2^2 = T.$$

Hence, we have

$$\sum_{k=1}^{T} \lambda_k \delta_k \leq \sqrt{T} \cdot \|x^* - x^{(1)}\|_2.$$

Since $x^{(k)+} \in P$, we have that

$$\delta_{k+1} = f(x^{(k+1)}) - f(x^*) \leq f(x^{(k)+}) - f(x^*) \leq \delta_k - \frac{1}{2L}\|\nabla f(x^{(k)})\|_2^2.$$

Hence, we have $\|\nabla f(x^{(k)})\|_2^2 \leq 2L(\delta_k - \delta_{k+1})$. This gives

$$\sum_{k=1}^{T} \frac{\delta_k}{\sqrt{\delta_k - \delta_{k+1}}} \leq \sqrt{2LT} \cdot \|x^* - x^{(1)}\|_2$$

for all $T$. Solving this recursion gives the result. □

**Exercise 8.3.2.** Solve the recursion omitted at the end of the proof.

Note that the proof above used only the fact that $\{x^{(1)}, x^{(k)+}, \sum_{s=1}^{k} \frac{\nabla f(x^{(s)})}{\|\nabla f(x^{(s)})\|}\} \subset P$. Therefore, one can put extra vectors in $P$ to obtain extra features. For example, if we use the subspace

$$P = x^{(k)} + \text{span}(x^{(k)} - x^{(1)}, \nabla f(x^{(k)}), \sum_{s=1}^{k} \frac{\nabla f(x^{(s)})}{\|\nabla f(x^{(s)})\|}, x^{(k)} - x^{(k-1)}),$$

then one can prove that this algorithm is equivalently to conjugate gradient when $f$ is a quadratic function.

## 8.4 Accelerated Gradient Descent

### Gradient Mapping

To give a general version of acceleration, we consider problem of the form

$$\min_x \phi(x) \stackrel{\text{def}}{=} f(x) + h(x)$$

where $f(x)$ is strongly convex and smooth and $h(x)$ is convex. We assume that we access the function $f$ and $h$ differently via:

1. Let $\mathcal{T}_f$ be the cost of computing $\nabla f(x)$.
2. Let $\mathcal{T}_{h,\lambda}$ be the cost of minimizing $h(x) + \frac{\lambda}{2}\|x - c\|_2^2$ exactly.

The idea is to move whatever we can optimize in $\phi$ to $h$ and hopefully this makes the remaining part of $\phi$, $f$, as smooth and strongly convex as possible. To make the statement general, we only assume $h$ is convex and hence $h$ may not be differentiable. To handle this issue, we need to define an approximate derivative of $h$ that we can compute.

**Definition 8.4.1.** We define the gradient step

$$p_x = \text{argmin}_y f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|^2 + h(y)$$

and the gradient mapping

$$g_x = L(x - p_{x,\gamma}).$$

Note that if $h = 0$, then $p_x = x - \frac{1}{L}\nabla f(x)$ and $g_x = \nabla f(x)$. In general, if $\phi \in \mathcal{C}^2$, then we have that

$$p_{x,\gamma} = x - \frac{1}{L}\nabla\phi(x) + O(\frac{1}{L^2}).$$

Therefore, we have that $g_x = \nabla\phi(x) + O(\frac{1}{L})$. Hence, the gradient mapping is an approximation of the gradient of $\phi$ that is computable in time $\mathcal{T}_g + \mathcal{T}_{h,L}$.

The key lemma we use here is that $\phi$ satisfies a lower bound defining using $g_x$. Ideally, we would love to get a lower bound as follows:

$$\phi(z) \geq \phi(x) + g_x^\top (z - x) + \frac{\mu}{2}\|z - x\|_2^2.$$

But it is WRONG. If that was true for all $z$, then we would have $g_x = \nabla\phi(x)$. However, if $\phi \in \mathcal{C}^2$ is $\mu$ strongly convex, then we have

$$\begin{aligned}\phi(z) &\geq \phi(x) + \nabla\phi(x)^\top (z - x) + \frac{\mu}{2}\|z - x\|_2^2 \\ &\geq \phi(x - \frac{1}{L}\nabla\phi(x)) + \frac{1}{2L}\|\nabla\phi(x)\|^2 + \nabla\phi(x)^\top (z - x) + \frac{\mu}{2}\|z - x\|_2^2. \qquad (8.4.1)\end{aligned}$$

It turns out that this is true and is exactly what we need for proving gradient descent, mirror descent and accelerated gradient descent.

**Theorem 8.4.2.** *Given $\phi = f + h$. Suppose that $f$ is $\mu$ strongly convex with $L$-Lipschitz gradient. Then, for any $z$, we have that*

$$\phi(z) \geq \phi(p_x) + g_x^\top (z - x) + \frac{1}{2L} \|g_x\|_2^2 + \frac{\mu}{2} \|z - x\|_2^2 .$$

*Proof.* To prove this, naturally, we want to follow the calculation in (8.4.1) and this corresponds to the following:

$$\begin{aligned}
\phi(z) - \frac{\mu}{2} \|z - x\|_2^2 &\geq f(x) + \nabla f(x)^\top (z - x) + h(z) \\
&= f(x) + \nabla f(x)^\top (p_x - x) + \nabla f(x)^\top (z - p_x) + h(z) \\
&\geq f(p_x) - \frac{L}{2} \|p_x - x\|^2 + \nabla f(x)^\top (z - p_x) + h(z) \\
&= \phi(p_x) - \frac{1}{2L} \|g_x\|^2 + \nabla f(x)^\top (z - p_x) + h(z) - h(p_x)
\end{aligned}$$

where we used $\text{Lip}\nabla f \leq L$. Unfortunately, this is not what we needed. Comparing this to what we need, it suffices to prove that

$$-\frac{1}{2L} \|g_x\|^2 + \nabla f(x)^\top (z - p_x) + h(z) - h(p_x) \geq g_x^\top (z - x) + \frac{1}{2L} \|g_x\|_2^2 .$$

Let $\overline{f}(y) = f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2} \|y - x\|_2^2$, then this is same as

$$\nabla \overline{f}(p_{x,\gamma})^\top (z - p_x) + h(z) - h(p_x) \geq 0 \tag{8.4.2}$$

Let $p_t = p_x + t(z - p_x)$. Using that $p_0$ is the minimizer of $\overline{f} + h$, we have that

$$\overline{f}(p_0) + h(p_0) \leq \overline{f}(p_t) + h(p_t) \leq \overline{f}(p_0) + \nabla \overline{f}(p_0)^\top (p_t - p_0) + \frac{L}{2} \|p_t - p_0\|^2 + h(p_t).$$

Hence, we have that

$$\begin{aligned}
0 &\leq \nabla \overline{f}(p_0)^\top (p_t - p_0) + \frac{L}{2} \|p_t - p_0\|^2 + h(p_t) - h(p_0) \\
&= t \cdot \left(\nabla \overline{f}(p_0)^\top (z - p_0) + h(z) - h(p_0)\right) + \frac{Lt^2}{2} \|z - p_0\|^2 .
\end{aligned}$$

Taking $t \to 0^+$, we have (8.4.2). $\square$

The next lemma shows that $\|g_x\|_2 \leq 2G$ if $\phi$ is $G$-Lipschitz.

**Lemma 8.4.3.** *If $\phi$ is $G$-Lipschitz, then $\|g_x\|_2 \leq 2G$ for all $x$.*

*Proof.* By the definition of gradient mapping (namely, $p_x$ is the minimizer of a function), we have that

$$f(x) + \nabla f(x)^\top (p_x - x) + \frac{L}{2} \|p_x - x\|^2 + h(p_x) \leq f(x) + h(x).$$

Using $h(p_x) \geq h(x) + \nabla h(x)^\top (p_x - x)$, we have that

$$0 \geq \nabla \phi(x)^\top (p_x - x) + \frac{L}{2} \|p_x - x\|^2 \geq -G \|p_x - x\|_2 + \frac{L}{2} \|p_x - x\|^2 .$$

Hence, we have that $\|p_x - x\|_2 \leq \frac{2}{L} G$ and hence $\|g_x\|_2 \leq 2G$. $\square$

## Gradient Descent Using Gradient Mapping

Putting $z = x$ in Theorem 8.4.2, we have the following:

**Lemma 8.4.4** (Gradient Descent Lemma)**.** *We have that*

$$\phi(p_x) \leq \phi(x) - \frac{1}{2L} \|g_x\|_2^2 .$$

This shows that each step of the gradient step decreases the function value by $\frac{1}{2L}\|g_x\|_2^2$. Therefore, if the gradient is large, then we decrease the function value by a lot. On the other hand, Putting $z = x^*$ for Theorem 8.4.2 shows that

$$\phi(x^*) \geq \phi(p_x) + g_x^\top (x^* - x).$$

If the gradient is small and domain is bounded, this shows that we are close to the optimal. Combining these two facts, we can get the gradient descent.

## Mirror Descent Using Gradient Mapping

Consider the mirror descent as

$$x^{(k+1)} = x^{(k)} - \eta g_{x^{(k)}}. \tag{8.4.3}$$

The main difference between this and gradient descent is that we will take a larger step size $h$. To analyze the mirror descent, we use Theorem 8.4.2 and the convexity of $\phi$ to get that

$$\phi(\frac{1}{k}\sum_{i=1}^{k} p_{x^{(i)}}) - \phi(x^*) \leq \frac{1}{k}\sum_{i=1}^{k}(\phi(p_{x^{(i)}}) - \phi(x^*)) \leq \frac{1}{k}\sum_{i=1}^{k} g_{x^{(i)}}^\top (x^{(i)} - x^*). \tag{8.4.4}$$

Therefore, it suffices to upper bound $g_{x^{(i)}}^\top (x^{(i)} - x^*)$. The following lemma shows that if $g_{x^{(i)}}^\top (x^{(i)} - x^*)$ is large, then either the gradient is large or the distance to optimum moves a lot. It turns out this holds for any vector $g$, not necessarily an approximate gradient.

**Lemma 8.4.5** (Mirror Descent Lemma)**.** *Let $p = x - \eta g$. Then, we have that*

$$g^\top (x - u) = \frac{\eta}{2}\|g\|_2^2 + \frac{1}{2\eta}\left(\|x - u\|_2^2 - \|p - u\|_2^2\right)$$

*for any $u$.*

## Algorithm and Analysis

Recall Lemma 8.4.4 shows that if the gradient is large, gradient descent makes a large progress. On the other hand, if the gradient is small, (8.4.4) shows that mirror descent makes a large progress. Therefore, it is natural to combine two approaches.

**Algorithm 26:** `AGD`

```
Input: Initial point $x^{(1)} \in \mathbb{R}^n$.
$y^{(1)} \leftarrow x^{(1)}$, $z^{(1)} \leftarrow x^{(1)}$.
for $k = 1, 2, \cdots, T$ do
    $x^{(k+1)} \leftarrow \tau z^{(k)} + (1-\tau) y^{(k)}$.
    Perform a gradient step: $y^{(k+1)} = x^{(k+1)} - \frac{1}{L} g_{x^{(k+1)}}$.
    Perform a mirror step: $z^{(k+1)} = z^{(k)} - \eta g_{x^{(k+1)}}$.
end
Return $\frac{1}{T}\sum_{k=1}^{T} p_{x^{(k+1)}}$.
```

Note that if $\tau = 1$, the algorithm is simply mirror descent and if $\tau = 0$, the algorithm is gradient descent.

**Theorem 8.4.6.** *Setting $\frac{1-\tau}{\tau} = \eta L$ and $\eta = \frac{1}{\sqrt{\mu L}}$, we have that*

$$\phi(\frac{1}{T}\sum_{k=1}^{T} p_{x^{(k+1)}}) - \phi(x^*) \leq \frac{2}{T}\sqrt{\frac{L}{\mu}}\left(\phi(x^{(1)}) - \phi(x^*)\right).$$

*In particular, if we restart the algorithm every $4\sqrt{\frac{L}{\mu}}$ iterations, we can find $x$ such that*

$$\phi(x) - \phi(x^*) \leq 2(1 - \sqrt{\frac{\mu}{L}})^{\Omega(T)}\left(\phi(x^{(1)}) - \phi(x^*)\right)$$

*in $T$ steps. Furthermore, each step takes $\mathcal{T}_f + \mathcal{T}_{h,L}$*

*Proof.* Lemma 8.4.5 showed that

$$g_{x^{(k+1)}}^\top (z^{(k)} - x^*) \leq \frac{\eta}{2} \|g_{x^{(k+1)}}\|_2^2 + \frac{1}{2\eta}\left(\left\|z^{(k)} - x^*\right\|_2^2 - \left\|z^{(k+1)} - x^*\right\|_2^2\right) \tag{8.4.5}$$

This shows that if the mirror descent has large error $g_{x^{(k+1)}}^\top (z^{(k)} - x^*)$, then the gradient descent makes a large progress $(\frac{\eta}{2} \|g_{x^{(k+1)}}\|_2^2)$.

To make the left-hand side usable, note that $x^{(k+1)} = z^{(k)} + \frac{1-\tau}{\tau} \cdot (y^{(k)} - x^{(k+1)})$ and hence

$$\begin{aligned} g_{x^{(k+1)}}^\top (x^{(k+1)} - x^*) &= g_{x^{(k+1)}}^\top (z^{(k)} - x^*) + \frac{1-\tau}{\tau} \cdot g_{x^{(k+1)}}^\top (y^{(k)} - x^{(k+1)}) \\ &\leq g_{x^{(k+1)}}^\top (z^{(k)} - x^*) + \frac{1-\tau}{\tau}(\phi(y^{(k)}) - \phi(y^{(k+1)}) - \frac{1}{2L}\|g_{x^{(k+1)}}\|_2^2) \\ &\leq \frac{\eta}{2}\|g_{x^{(k+1)}}\|_2^2 + \frac{1}{2\eta}\left(\left\|z^{(k)} - x^*\right\|_2^2 - \left\|z^{(k+1)} - x^*\right\|_2^2\right) + \frac{1-\tau}{\tau}(\phi(y^{(k)}) - \phi(y^{(k+1)}) - \frac{1}{2L}\|g_{x^{(k+1)}}\|_2^2). \end{aligned}$$

where we used Theorem 8.4.2 in the middle and (8.4.5) at the end.

Now, we set $\frac{1-\tau}{\tau} = \eta L$ and get

$$g_{x^{(k+1)}}^\top (x^{(k+1)} - x^*) \leq \eta L(\phi(y^{(k)}) - \phi(y^{(k+1)})) + \frac{1}{2\eta}\left(\left\|z^{(k)} - x^*\right\|_2^2 - \left\|z^{(k+1)} - x^*\right\|_2^2\right).$$

Taking a sum on both side and let $\overline{x} = \frac{1}{T}\sum_{k=1}^T p_{x^{(k+1)}}$, we have that

$$\begin{aligned} \phi(\overline{x}) - \phi(x^*) &\leq \frac{1}{T}\sum_{k=1}^T (\phi(p_{x^{(k+1)}}) - \phi(x^*)) \\ &\leq \frac{1}{T}\sum_{k=1}^T g_{x^{(k+1)}}^\top (x^{(k+1)} - x^*) \\ &\leq \frac{\eta L}{T}\left(\phi(y^{(1)}) - \phi(y^{(T+1)})\right) + \frac{1}{2\eta T}\left\|z^{(1)} - x^*\right\|_2^2 \\ &\leq \left(\frac{\eta L}{T} + \frac{1}{2\eta T \mu}\right)\left(\phi(x^{(1)}) - \phi(x^*)\right) \end{aligned}$$

The conclusion follows from our setting of $\eta$. □

## 8.5 Accelerated Coordinate Descent

Recall from Theorem 6.4.2 that if $\frac{\partial^2}{\partial x_i^2}\ell(x) \leq L_i$ for all $x$ and that $\ell$ is $\mu$ strongly convex, then we can find $x^{(k)}$ in $k$ coordinate steps such that

$$\mathbf{E}\ell(x^{(k)}) - \ell(x^*) \leq (1 - \frac{\mu}{L})^{\Omega(k)}(\ell(x^{(0)}) - f(x^*))$$

where $L = \sum_i L_i$. Now, the really fun part is here. Consider the function

$$\phi(x) = f(x) + h(x) \quad \text{with} \quad f(x) = \frac{\mu}{2}\|x\|_2^2 \text{ and } h(x) = \ell(x) - \frac{\mu}{2}\|x\|_2^2.$$

Since $f$ is $\mu + \frac{L}{n}$ smooth (YES!, I know this is also $\mu$ smooth) and $\mu$ strongly convex and since $h$ is convex, we apply Theorem 8.4.6 and get an algorithm that takes $O^*(\sqrt{\frac{L}{n\mu}})$ steps. Note that each step involves $\mathcal{T}_f + \mathcal{T}_{h,\mu+\frac{L}{n}}$.

Obviously, $\mathcal{T}_f = 0$. Next, note that $\mathcal{T}_{h,\mu+\frac{L}{n}}$ involves solving a problem of the form

$$\begin{aligned} y_x &= \operatorname{argmin}_y (\frac{\mu}{2} + \frac{L}{2n}) \|y-x\|^2 + (\ell(y) - \frac{\mu}{2}\|x\|^2) \\ &= \operatorname{argmin}_y \ell(y) - \mu y^\top x + \frac{L}{2n}\|y-x\|^2 . \end{aligned}$$

Now, we can apply Theorem 6.4.2 to solve this problem. It takes

$$O^*(\frac{L + (L/n)\cdot n}{L/n}) = O^*(n) \text{ coordinate steps.}$$

Therefore, in total it takes

$$O^*(\sqrt{\frac{L}{n\mu}}) \cdot O^*(n) = O^*(\sqrt{\frac{Ln}{\mu}}) \text{ coordinate steps.}$$

Hence, we have the following theorem

**Theorem 8.5.1** (Accelerated Coordinate Descent Convergence)**.** *Given an $\mu$ strongly-convex function $\ell$. Suppose that $\frac{\partial^2}{\partial x_i^2}\ell(x) \leq L_i$ for all $x$ and let $L = \sum L_i$. We can minimize $\ell$ in* $O^*(\sqrt{\frac{Ln}{\mu}})$ coordinate steps.

*Remark* 8.5.2. It is known how to do it in $O^*(\sum_i \sqrt{\frac{L_i}{\mu}})$ steps [3].

## ■ 8.6 Accelerated Stochastic Descent

Recall from Theorem 6.3.7 that if $\nabla^2 \ell_i(x) \preceq L$ for all $x$ and $i$ and that $\ell(x) = \frac{1}{n}\sum_{i=1}^n \ell_i(x)$ is $\mu$ strongly convex, then we can find $x$ in $O((n + \frac{L}{\mu})\log(\frac{1}{\epsilon}))$ stochastic steps such that

$$\mathbf{E}\ell(x) - \ell(x^*) \leq \epsilon(\ell(x^{(0)}) - \ell(x^*)).$$

Similar to the coordinate descent, we can accelerate it using the accelerated gradient descent (Theorem 8.4.6). To apply Theorem 8.4.6, we consider the function

$$\phi(x) = f(x) + h(x) \quad \text{with} \quad f(x) = \frac{\mu}{2}\|x\|_2^2 \text{ and } h(x) = \ell(x) - \frac{\mu}{2}\|x\|_2^2 .$$

Since $f$ is $\mu + \frac{L}{n}$ smooth (YES!, I know this is also $\mu$ smooth) and $\mu$ strongly convex and since $h$ is convex, we apply Theorem 8.4.6 and get an algorithm that takes $O^*(1+\sqrt{\frac{L}{n\mu}})$ steps. Note that each step involves $\mathcal{T}_f + \mathcal{T}_{h,\mu+\frac{L}{n}}$. Obviously, $\mathcal{T}_f = 0$. Next, note that $\mathcal{T}_{h,\mu+\frac{L}{n}}$ involves solving a problem of the form

$$\begin{aligned} y_x &= \operatorname{argmin}_y (\frac{\mu}{2} + \frac{L}{2n}) \|y-x\|^2 + (\ell(y) - \frac{\mu}{2}\|x\|^2) \\ &= \operatorname{argmin}_y \ell(y) - \mu y^\top x + \frac{L}{2n}\|y-x\|^2 \\ &= \operatorname{argmin}_y \frac{1}{n}\sum_i (\ell_i(y) + \frac{L}{2n}\|y-x\|^2 - \mu y^\top x) \end{aligned}$$

Now, we can apply Theorem 6.3.7 to solve this problem. It takes

$$O^*(n + \frac{L + \frac{L}{n}}{\mu + \frac{L}{n}}) = O^*(n)$$

Therefore, in total it takes

$$O^*(1 + \sqrt{\frac{L}{n\mu}}) \cdot O^*(n) = O^*(n + \sqrt{n\frac{L}{\mu}}).$$

**Theorem 8.6.1.** *Given a convex function $\ell = \frac{1}{n}\sum \ell_i$. Suppose that $\nabla^2 \ell_i(x) \preceq L$ for all $i$ and $x$ and that $\ell$ is $\mu$ strongly convex. Suppose we can compute $\nabla \ell_i$ in $O(1)$ time. We have an algorithm that outputs an $x$ such that*

$$\mathbf{E}\ell(x) - \ell(x^*) \leq \varepsilon(\ell(x^{(0)}) - \ell(x^*))$$

*in* $O^*(n + \sqrt{n\frac{L}{\mu}})$ *stochastic steps.*

# Chapter 9

# Expansion

## 9.1 Ordinary Differential Equations

We consider the first-order ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = F(x(t),t), \quad \forall 0 \le t \le T,$$
$$x(0) = v.$$

where $F : \mathbb{R}^{d+1} \to \mathbb{R}$. The basic idea to solve it comes from the Picard-Lindelöf theorem, a constructive existence proof for a large class of ODEs.

### Picard-Lindelöf theorem

In general, we can rewrite the first-order ODE as an integral equation

$$x(t) = v + \int_0^t F(x(s),s)\mathrm{d}s \quad \text{for all } 0 \le t \le T.$$

To simplify the notation, we use $\mathcal{C}([0,T],\mathbb{R}^d)$ to denote $\mathbb{R}^d$-valued functions on $[0,T]$. We define the operator $\mathcal{T}$ from $\mathcal{C}([0,T],\mathbb{R}^d)$ to $\mathcal{C}([0,T],\mathbb{R}^d)$ by

$$\mathcal{T}(x)(t) = v + \int_0^t F(x(s),s)\mathrm{d}s \quad \text{for all } 0 \le t \le T. \tag{9.1.1}$$

Therefore, the integral equation is simply $x = \mathcal{T}(x)$. Let $\mathcal{T}^{\circ j}$ be the composition of $j$ many $\mathcal{T}$, i.e.,

$$\mathcal{T}^{\circ j}(x) = \underbrace{\mathcal{T}(\mathcal{T}(\cdots\mathcal{T}}_{j \text{ many } \mathcal{T}}(x)\cdots)).$$

The Banach fixed point theorem says that the integral equation $x = \mathcal{T}(x)$ has a unique solution if there is a norm, and $j \in \mathbb{N}$ such that the map $\mathcal{T}^{\circ j}$ has Lipschitz constant less than 1.

The Picard-Lindelöf theorem shows that if $F$ is Lipschitz in $x$, then the map $\mathcal{T}^{\circ j}$ has Lipschitz constant less than 1 for some positive integer $j$.

**Theorem 9.1.1.** *Given any norm $\|\cdot\|$ on $\mathbb{R}^d$. Let $L$ be the Lipschitz constant of $F$ in $x$, i.e.*

$$\|F(x,s) - F(y,s)\| \le L\|x-y\| \quad \text{for all } x,y \in \mathbb{R}^d, s \in [0,T].$$

*For any $x \in \mathcal{C}([0,T],\mathbb{R}^d)$, we define the corresponding norm*

$$\|x\| \overset{\text{def}}{=} \max_{0\le t\le T} \|x(t)\|.$$

*Then, the Lipschitz constant of $\mathcal{T}^{\circ j}$ in this norm is upper bounded by $(LT)^j/j!$.*

*Proof.* We prove this lemma with a stronger induction hypothesis

$$\|(\mathcal{T}^{\circ j}x)(h) - (\mathcal{T}^{\circ j}y)(h)\| \leq \frac{L^j h^j}{j!}\|x-y\| \quad \text{for all } 0 \leq h \leq T.$$

The base case $j = 0$ is trivial. For the inductive step,

$$\begin{aligned}
\|\mathcal{T}^{\circ j}x(h) - \mathcal{T}^{\circ j}y(h)\| &= \|\mathcal{T}\mathcal{T}^{\circ(j-1)}x(h) - \mathcal{T}\mathcal{T}^{\circ(j-1)}y(h)\| \\
&\leq \int_0^h \|F(\mathcal{T}^{\circ(j-1)}x(s), s) - F(\mathcal{T}^{\circ(j-1)}y(s), s)\|\mathrm{d}s \\
&\leq L\int_0^h \|\mathcal{T}^{\circ(j-1)}x(s) - \mathcal{T}^{\circ(j-1)}y(s)\|\mathrm{d}s && \text{by } f \text{ is } L \text{ Lipschitz} \\
&\leq L\int_0^h \frac{L^{j-1}s^{j-1}}{(j-1)!}\|x-y\|\mathrm{d}s && \text{by the induction hypothesis} \\
&= \frac{L^j}{j!}h^j\|x-y\|.
\end{aligned}$$

This completes the proof. □

To make the Picard-Lindelöf theorem algorithmic, we have to decide how to represent a function in $\mathcal{C}([0,T],\mathbb{R}^d)$. One standard way is to use a polynomial $p_i(t)$ in $t$ for each coordinate $i \in [d]$. More generally, suppose there is a basis $\{\varphi_j\}_{j=1}^D \subset \mathcal{C}([0,T],\mathbb{R})$ such that for all $i \in [d]$, $\frac{\mathrm{d}x_i}{\mathrm{d}t}$ is approximated by some linear combination of $\varphi_j(t)$. For example, if $\varphi_j(t) = t^{j-1}$ for $j \in [D]$, then our assumption is simply saying $\frac{\mathrm{d}x_i}{\mathrm{d}t}$ is approximated by a degree $D-1$ polynomial. Other possible basis are piecewise-polynomial and Fourier series. By Gram-Schmidt orthogonalization, we can always pick points $\{c_i\}_{j=1}^D$ such that

$$\varphi_j(c_i) = \delta_{i,j} \quad \text{for } i,j \in [D].$$

The benefit of such a basis is that for any $f \in \text{span}(\varphi_j)$, we have that $f(t) = \sum_{j=1}^D f(c_j)\varphi_j(t)$.

For polynomials, we can use the basis consisting of the Lagrange polynomials:

$$\varphi_j(t) = \prod_{i\in[D]\setminus\{j\}} \frac{t-c_i}{c_j-c_i} \quad \text{for } j \in [D].$$

The only assumption we make on the basis is that it is bounded in the following sense.

**Definition 9.1.2.** Given a $D$ dimensional subspace $\mathcal{V} \subset \mathcal{C}([0,T],\mathbb{R})$ and node points $\{c_j\}_{j=1}^D \subset [0,T]$. For any $\gamma_\varphi \geq 1$, we say that a basis $\{\varphi_j\}_{j=1}^D \subset \mathcal{V}$ is $\gamma_\varphi$-bounded if $\varphi_j(c_i) = \delta_{i,j}$ and we have

$$\sum_{j=1}^D \left|\int_0^t \varphi_j(s)\mathrm{d}s\right| \leq \gamma_\varphi T \quad \text{for } t \in [0,T].$$

Note that if the constant function $1 \in \mathcal{V}$, then we have

$$1 = \sum_{j=1}^D 1(c_j)\varphi_j(t) = \sum_{j=1}^D \varphi_j(t).$$

Hence,

$$T = \int_0^T 1\mathrm{d}s \leq \sum_{j=1}^D \left|\int_0^T \varphi_j(s)\mathrm{d}s\right| \leq \gamma_\varphi T.$$

Therefore, $\gamma_\varphi \geq 1$ for most elements. Later we will see that for the space of low degree polynomials or piece-wise low-degree polynomials, there is a basis on the Chebyshev nodes that is $O(1)$-bounded.

Assuming that $\frac{dx}{dt}$ can be approximated by some element in $\mathcal{V}$, we have that

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) \sim \sum_{j=1}^D \frac{\mathrm{d}x}{\mathrm{d}t}(c_j)\varphi_j(t) = \sum_{j=1}^D F(x(c_j), c_j)\varphi_j(t).$$

Integrating both sides, we have

$$x(t) \approx v + \int_0^t \sum_{j=1}^{D} F(x(c_j), c_j)\varphi_j(s)\mathrm{d}s. \tag{9.1.2}$$

This suggests the following operator from $\mathcal{C}([0,T],\mathbb{R}^d)$ to $\mathcal{C}([0,T],\mathbb{R}^d)$:

$$\mathcal{T}_\varphi(x)(t)_i = v_i + \int_0^t \sum_{j=1}^{D} F(x(c_j), c_j)_i\varphi_j(s)\mathrm{d}s \quad \text{for } i \in [d]. \tag{9.1.3}$$

Equation (9.1.2) can be written as $x \approx \mathcal{T}_\varphi(x)$. To find $x$ that satisfies this, naturally, one can apply a fix point iteration and the resulting algorithm is called the collocation method.

## 9.2 Collocation Method

From the definition of (9.1.3), we note that $\mathcal{T}_\varphi(x)$ depends only on $x(t)$ at $t = c_j$. Therefore, we can only need to calculate $\mathcal{T}_\varphi(x)(t)$ at $t = c_j$. To simplify the notation, for any $x \in \mathcal{C}([0,T],\mathbb{R}^d)$, we define a corresponding matrix $[x] \in \mathbb{R}^{d\times D}$ by $[x]_{i,j} = x_i(c_j)$. For any $d \times D$ matrix $X$, we define $F(X,c)$ as an $d \times D$ matrix

$$F(X,c)_{i,j} = F(X_{*,j}, c_j)_i. \tag{9.2.1}$$

where $X_{*,j}$ is the $j$-th column of $X$. We also define $A_\varphi$ as a $D \times D$ matrix

$$(A_\varphi)_{i,j} = \int_0^{c_j} \varphi_i(s)\mathrm{d}s. \tag{9.2.2}$$

Note that $A_\phi$ can be pre-computed. By inspecting the definition of (9.1.3), (9.2.1) and (9.2.2), we have that

$$[\mathcal{T}_\varphi(x)] = v \cdot 1_D^\top + F([x], c)A_\varphi$$

where $1_D$ is a column of all 1 vector of length $D$. Hence, we can apply the map $\mathcal{T}_\varphi$ by simply multiply $F([x],c)$ by a pre-compute $D \times D$ matrix $A_\varphi$. For the basis we consider here, each iteration takes only $\tilde{O}(dD)$ which is nearly linear in the size of our representation of the solution.

**Algorithm 27:** `CollocationMethod`}

```
Input: $F, v, T, \varphi, c$.
Let $N = \lceil \log\left(\frac{T}{\epsilon}\max_{s\in[0,T]}\|F(v,s)\|\right)\rceil$. Let $(A_\varphi)_{i,j} = \int_0^{c_j}\varphi_i(s)\mathrm{d}s$.
$X^{(0)} \leftarrow v \cdot 1_D^\top$.
For $j = 1, 2, \cdots, N-1$:
    $X^{(j)} \leftarrow v \cdot 1_D^\top + F(X^{(j-1)}, c)A_\varphi$.
$x^{(N)}(t) \leftarrow v + \int_0^t \sum_{i=1}^D F(X_{*,i}^{(N)}, c_i)\varphi_i(s)\mathrm{d}s$.
return $x^{(N)}$.
```

We state our guarantee for a first-order ODE. It can be extended to a $k$'th order ODE.

**Theorem 9.2.1.** *Let $x^*(t)$ be the solution of an $d$ dimensional ODE*

$$x(0) = v, \frac{\mathrm{d}x(t)}{\mathrm{d}t} = F(x(t), t) \quad \text{for all } 0 \le t \le T.$$

*We are given a $D$ dimensional subspace $\mathcal{V} \subset \mathcal{C}([0,T],\mathbb{R})$, node points $\{c_j\}_{j=1}^D \subset [0,T]$ and a $\gamma_\varphi$ bounded basis $\{\varphi_j\}_{j=1}^D \subset \mathcal{V}$ (Definition 9.1.2). Let $L$ and $\epsilon > 0$ be such that there exists a function $q \in \mathcal{V}$ such that*

$$\left\|q(t) - \frac{\mathrm{d}}{\mathrm{d}t}x^*(t)\right\| \le \frac{\epsilon}{T}, \forall t \in [0,T]$$

*and for any $y, z \in \mathbb{R}^d$,*

$$\|F(y,t) - F(z,t)\| \le L\|y - z\|, \forall t \in [0,T].$$

*Assume $\gamma_\varphi LT \le 1/2$. Then Algorithm* CollocationMethod *outputs a function $x^{(N)} \in \mathcal{V}$ such that*

$$\max_{t\in[0,T]} \|x^{(N)}(t) - x^*(t)\| \le 20\gamma_\varphi \epsilon$$

*using $O\left(D \log\left(\frac{T}{\epsilon} \max_{s\in[0,T]} \|F(v,s)\|\right)\right)$ evaluations of $F$.*

## Proof for first-order ODE

First, we bound the Lipschitz constant of the map $\mathcal{T}_\varphi$. Unlike in the Picard-Lindelöf theorem, we are not able to get an improved bound of the Lipschitz constant by composing many copies of $\mathcal{T}_\varphi$. Fortunately, the Lipschitz constant of the map $\mathcal{T}_\varphi$ can be bounded.

**Lemma 9.2.2.** *Given any norm $\|\cdot\|$ on $\mathbb{R}^d$. Let $L$ be the Lipschitz constant of $F$ in $x$, i.e.*

$$\|F(x,s) - F(y,s)\| \le L\|x-y\| \quad \textit{for all } x,y \in \mathbb{R}^d, s\in[0,T].$$

*Then, the Lipschitz constant of $\mathcal{T}_\varphi$ in this norm is upper bounded by $\gamma_\varphi LT$.*

*Proof.* For any $0 \le t \le T$,

$$\begin{aligned}\|\mathcal{T}_\varphi(x)(t) - \mathcal{T}_\varphi(y)(t)\| &= \left\|\int_0^t \sum_{j=1}^D F(x(c_j),c_j)\varphi_j(s)\mathrm{d}s - \int_0^t \sum_{j=1}^D F(y(c_j),c_j)\varphi_j(s)\mathrm{d}s\right\| \\ &\le \sum_{j=1}^D \left|\int_0^t \varphi_j(s)\mathrm{d}s\right| \cdot \max_{t\in[0,T]} \|F(x(t),t) - F(y(t),t)\| \\ &\le \gamma_\varphi LT \cdot \max_{t\in[0,T]} \|x(t)-y(t)\| \\ &\le \gamma_\varphi LT\|x-y\|.\end{aligned}$$

where the third step follows using $\sum_{j=1}^D |\int_0^t \varphi_j(s)\mathrm{d}s| \le \gamma_\varphi T$ for all $0\le t\le T$. □

For the rest of the proof, let $x_\varphi^*$ denote the fixed point of $\mathcal{T}_\varphi$, i.e., $\mathcal{T}_\varphi(x_\varphi^*) = x_\varphi^*$. The Banach fixed point theorem and Lemma 9.2.2 imply that $x_\varphi^*$ exists and is unique if $T \le \frac{1}{L\gamma_\varphi}$.

Let $x^*$ denote the solution of the ODE, i.e., the fixed point of $\mathcal{T}$, with $\mathcal{T}(x^*) = x^*$. Let $x^{(0)}$ denote the initial solution given by $x^{(0)}(t) = v$ and $x^{(N)}$ denote the solution obtained by applying operator $\mathcal{T}_\varphi$ for $N$ times. Note that $x^{(N)}(t)$ is the output of CollocationMethod.

Let $q \in \mathcal{V}$ denote an approximation of $\frac{\mathrm{d}}{\mathrm{d}t}x^*$ such that

$$\left\|q(t) - \frac{\mathrm{d}}{\mathrm{d}t}x^*(t)\right\| \le \frac{\epsilon}{T}. \forall t\in[0,T]$$

The next lemma summarizes how these objects are related and will allow us to prove the main guarantee for first-order ODEs.

**Lemma 9.2.3.** *Let $L^{(j)}$ be the Lipschitz constant of the map $\mathcal{T}_\varphi^{\circ j}$. Assume that $L^{(N)} \le 1/2$. Then, we have*

$$\begin{aligned}\|x^{(N)} - x^*\| &\le L^{(N)}\|x^{(0)} - x^*\| + 2\|x_\varphi^* - x^*\|, &(9.2.3)\\ \|x_\varphi^* - x^*\| &\le 2\cdot\|\mathcal{T}_\varphi^{\circ N}(x^*) - x^*\|, &(9.2.4)\\ \|x^* - \mathcal{T}_\varphi^{\circ N}(x^*)\| &\le \sum_{i=0}^{N-1} L^{(i)}\cdot\|x^* - \mathcal{T}_\varphi(x^*)\|, &(9.2.5)\\ \|x^* - \mathcal{T}_\varphi(x^*)\| &\le 2\gamma_\varphi\cdot\epsilon. &(9.2.6)\end{aligned}$$

*Proof.* We prove the claims in order. For the first claim,

$$\begin{aligned}
\|x^{(N)} - x^*\| &\le \|x^{(N)} - x^*_\varphi\| + \|x^*_\varphi - x^*\| && \text{by triangle inequality}\\
&= \|\mathcal{T}_\varphi^{\circ N}(x^{(0)}) - \mathcal{T}_\varphi^{\circ N}(x^*_\varphi)\| + \|x^*_\varphi - x^*\|\\
&\le L^{(N)}\|x^{(0)} - x^*_\varphi\| + \|x^*_\varphi - x^*\|\\
&\le L^{(N)}\|x^{(0)} - x^*\| + L^{(N)}\|x^* - x^*_\varphi\| + \|x^*_\varphi - x^*\|\\
&\le L^{(N)}\|x^{(0)} - x^*\| + 2\|x^*_\varphi - x^*\|
\end{aligned}$$

where the last step follows by $L^{(N)} \le 1$.

For the second claim,

$$\begin{aligned}
\|x^*_\varphi - x^*\| &= \|\mathcal{T}_\varphi^{\circ N}(x^*_\varphi) - x^*\| && \text{by } x^*_\varphi = \mathcal{T}_\varphi^{\circ N}(x^*_\varphi)\\
&\le \|\mathcal{T}_\varphi^{\circ N}(x^*_\varphi) - \mathcal{T}_\varphi^{\circ N}(x^*)\| + \|\mathcal{T}_\varphi^{\circ N}(x^*) - x^*\| && \text{by triangle inequality}\\
&\le L^{(N)} \cdot \|x^*_\varphi - x^*\| + \|\mathcal{T}_\varphi^{\circ N}(x^*) - x^*\| && \text{by the definition of } L^{(N)}\\
&\le \frac{1}{2}\|x^*_\varphi - x^*\| + \|\mathcal{T}_\varphi^{\circ N}(x^*) - x^*\| && \text{by } L^{(N)} \le 1/2
\end{aligned}$$

For the third claim,

$$\begin{aligned}
\|x^* - \mathcal{T}_\varphi^{\circ N}(x^*)\| &\le \sum_{i=0}^{N-1} \|\mathcal{T}_\varphi^{\circ i}(x^*) - \mathcal{T}_\varphi^{\circ (i+1)}(x^*)\|\\
&\le \sum_{i=0}^{N-1} L^{(i)} \cdot \|x^* - \mathcal{T}_\varphi(x^*)\|.
\end{aligned}$$

For the last claim,

$$\begin{aligned}
\|x^*(t) - \mathcal{T}_\varphi(x^*)(t)\| &= \|\mathcal{T}(x^*)(t) - \mathcal{T}_\varphi(x^*)(t)\|\\
&= \left\|\int_0^t F(x^*(s), s)\mathrm{d}s - \int_0^t \sum_{j=1}^D F(x^*(c_j), c_j)\varphi_j(s)\mathrm{d}s\right\|\\
&= \left\|\int_0^t \frac{\mathrm{d}}{\mathrm{d}t}x^*(s)\mathrm{d}s - \int_0^t \sum_{j=1}^D \frac{\mathrm{d}}{\mathrm{d}t}x^*(c_j)\varphi_j(s)\mathrm{d}s\right\|\\
&\le \left\|\int_0^t (\frac{\mathrm{d}}{\mathrm{d}t}x^*(s) - q(s))\mathrm{d}s - \int_0^t \sum_{j=1}^D (\frac{\mathrm{d}}{\mathrm{d}t}x^*(c_j) - q(c_j))\varphi_j(s)\mathrm{d}s\right\|\\
&\quad + \left\|\int_0^t q(s)\mathrm{d}s - \int_0^t \sum_{j=1}^D q(c_j)\varphi_j(s)\mathrm{d}s\right\|\\
&\le \int_0^t \left\|\frac{\mathrm{d}}{\mathrm{d}t}x^*(s) - q(s)\right\|\mathrm{d}s + \sum_{j=1}^D \left\|\frac{\mathrm{d}}{\mathrm{d}t}x^*(c_j) - q(c_j)\right\| \left|\int_0^t \varphi_j(s)\mathrm{d}s\right| + 0\\
&\le (1 + \gamma_\varphi) \cdot \epsilon + 0
\end{aligned}$$

where the first step uses $\mathcal{T}(x^*) = x^*$, the second step follows by the definition of $\mathcal{T}$ and $\mathcal{T}_\varphi$, the third step follows by $x^*(t)$ is the solution of ODE, the fourth step follows by triangle inequality, the second last step uses $q \in \mathcal{V}$, and the last step follows from the assumption $\|\frac{\mathrm{d}}{\mathrm{d}t}x^* - q\| \le \frac{\epsilon}{T}$ and the definition of $\gamma_\varphi$. □

Now, we are ready to prove Theorem 9.2.1.

*Proof.* Using Lemma 9.2.3, we have

$$\begin{aligned}
\|x^{(N)} - x^*\| &\leq L^{(N)}\|x^{(0)} - x^*\| + 2\|x^*_\varphi - x^*\| && \text{by Eq. (9.2.3)}\\
&\leq L^{(N)}\|x^{(0)} - x^*\| + 4\|\mathcal{T}_\varphi^{\circ N}(x^*) - x^*\| && \text{by Eq. (9.2.4)}\\
&\leq L^{(N)}\|x^{(0)} - x^*\| + 4\sum_{i=0}^{N-1} L^{(i)} \cdot \|x^* - \mathcal{T}_\varphi(x^*)\| && \text{by Eq. (9.2.5)}\\
&\leq L^{(N)}\|x^{(0)} - x^*\| + 8\sum_{i=0}^{N-1} L^{(i)} \cdot \gamma_\varphi \cdot \epsilon. && \text{by Eq. (9.2.6)}
\end{aligned}$$

Using the assumption that $\gamma_\varphi LT \leq \frac{1}{2}$, Lemma 9.2.2 shows that $L^{(1)} \leq \frac{1}{2}$ and hence $L^{(j)} \leq \frac{1}{2^j}$. Therefore, we have

$$\|x^{(N)} - x^*\| \leq \frac{1}{2^N}\|x^{(0)} - x^*\| + 16\gamma_\varphi \cdot \epsilon = \frac{1}{2^N}\|x^* - x^*(0)\| + 16\gamma_\varphi \cdot \epsilon \tag{9.2.7}$$

To bound $\|x^* - x^*(0)\|$, for any $0 \leq t \leq T$

$$x^*(t) = x^*(0) + \int_0^t F(x^*(s), s)\mathrm{d}s.$$

Hence, we have that

$$\begin{aligned}
\|x^*(t) - x^*(0)\| &\leq \left\|\int_0^T F(x^*(0), s)\mathrm{d}s\right\| + \left\|\int_0^t \left(F(x^*(s), s) - F(x^*(0), s)\right)\mathrm{d}s\right\|\\
&\leq \left\|\int_0^T F(x^*(0), s)\mathrm{d}s\right\| + L\int_0^t \|x^*(s) - x^*(0)\|\mathrm{d}s.
\end{aligned}$$

Solving this integral inequality (see Lemma 9.2.4), we have that

$$\|x^*(t) - x^*(0)\| \leq e^{Lt}\left\|\int_0^T F(x^*(0), s)\mathrm{d}s\right\|.$$

Now, we use $LT \leq \frac{1}{2}$ and get

$$\|x^*(t) - x^*(0)\| \leq 2\left\|\int_0^T F(x^*(0), s)\mathrm{d}s\right\|.$$

Picking $N = \left\lceil \log_2\left(\frac{T}{\epsilon}\max_{s\in[0,T]}\|F(x^*(0), s)\|\right)\right\rceil$, (9.2.7) shows that the error is less than $20\gamma_\varphi\epsilon$. $\square$

**Lemma 9.2.4.** *Given a continuous function $v(t)$ and positive scalars $\beta, \gamma$ such that*

$$0 \leq v(t) \leq \beta + \gamma\int_0^t v(s)\mathrm{d}s.$$

*We have that $v(t) \leq \beta e^{\gamma t}$ for all $t \geq 0$.*

**Exercise 9.2.5.** Prove the above lemma.

[1] Radosław Adamczak, Alexander Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *Journal of the American Mathematical Society*, 23(2):535–561, 2010.

[2] Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. In *Advances in Neural Information Processing Systems*, pages 1614–1622, 2016.

[3] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119, 2016.

[4] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. In *STOC*, pages 156–163, 1991.

[5] Shiri Artstein-Avidan and Vitali Milman. The concept of duality in convex analysis, and the characterization of the legendre transform. *Annals of mathematics*, pages 661–674, 2009.

[6] David S Atkinson and Pravin M Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69(1-3):1–43, 1995.

[7] Amitabh Basu and Timm Oertel. Centerpoints: A link between optimization and convex geometry. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 14–25. Springer, 2016.

[8] Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In *Conference on Learning Theory*, pages 240–265, 2015.

[9] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.

[10] Sébastien Bubeck, Ronen Eldan, and Yin Tat Lee. Kernel-based methods for bandit convex optimization. *arXiv preprint arXiv:1607.03084*, 2016.

[11] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *arXiv preprint arXiv:1710.11606*, 2017.

[12] Michael B Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 278–287. SIAM, 2016.

[13] Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 181–190. ACM, 2015.

[14] Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *arXiv preprint arXiv:1810.07896*, 2018.

[15] B. Cousins and S. Vempala. Bypassing KLS: Gaussian cooling and an $O^*(n^3)$ volume algorithm. In *STOC*, pages 539–548, 2015.

[16] Dmitriy Drusvyatskiy, Maryam Fazel, and Scott Roy. An optimal first order method based on optimal quadratic averaging. *arXiv preprint arXiv:1604.06543*, 2016.

[17] M. E. Dyer and A. M. Frieze. Computing the volume of a convex body: a case where randomness provably helps. In *Proc. of AMS Symposium on Probabilistic Combinatorics and Its Applications*, pages 123–170, 1991.

[18] M. E. Dyer, A. M. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. In *STOC*, pages 375–381, 1989.

[19] Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for stochastic convex optimization. *CoRR*, abs/1512.09170, 2015.

[20] R Fletcher. Practical methods of optimization. 1987.

[21] Roger Fletcher. A new variational result for quasi-newton formulae. *SIAM Journal on Optimization*, 1(1):18–21, 1991.

[22] Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning*, pages 2540–2548, 2015.

[23] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Algorithms and Combinatorics, 1988.

[24] Adam Tauman Kalai and Santosh Vempala. Simulated annealing for convex optimization. *Math. Oper. Res.*, 31(2):253–266, February 2006.

[25] R. Kannan, L. Lovász, and M. Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures and Algorithms*, 11:1–50, 1997.

[26] Ravi Kannan, László Lovász, and Miklós Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete & Computational Geometry*, 13(1):541–559, 1995.

[27] Ravindran Kannan and Hariharan Narayanan. Random walks on polytopes and an affine interior point method for linear programming. *Mathematics of Operations Research*, 37(1):1–20, 2012.

[28] Ravindran Kannan and Santosh Vempala. Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26:95–135, 2017.

[29] Jonathan A Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 217–226. SIAM, 2014.

[30] L Khachiyan, S Tarasov, and E Erlich. The inscribed ellipsoid method. In *Soviet Math. Dokl*, volume 298, 1988.

[31] Leonid G Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.

[32] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 147–156. IEEE, 2013.

[33] Yin Tat Lee, Aaron Sidford, and Santosh S Vempala. Efficient convex optimization with membership oracles. *arXiv preprint arXiv:1706.07357*, 2017.

[34] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1049–1065. IEEE, 2015.

[35] Yin Tat Lee and Santosh S Vempala. Stochastic localization+ stieltjes barrier= tight bound for log-sobolev. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1122–1129. ACM, 2018.

[36] A Yu Levin. On an algorithm for the minimization of convex functions. In *Soviet Mathematics Doklady*, volume 160, pages 1244–1247, 1965.

[37] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

[38] L. Lovász. How to compute the volume? *Jber. d. Dt. Math.-Verein, Jubiläumstagung 1990*, pages 138–151, 1990.

[39] L. Lovász and M. Simonovits. Mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *ROCS*, pages 482–491, 1990.

[40] L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. In *Random Structures and Alg.*, volume 4, pages 359–412, 1993.

[41] L. Lovász and S. Vempala. Hit-and-run from a corner. *SIAM J. Computing*, 35:985–1005, 2006.

[42] L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Comput. Syst. Sci.*, 72(2):392–417, 2006.

[43] László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993.

[44] László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an o*(n4) volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006.

[45] László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007.

[46] Aleksander Madry. Fast approximation algorithms for cut-based problems in undirected graphs. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 245–254. IEEE, 2010.

[47] Vahab Mirrokni, Renato Paes Leme, Adrian Vladu, and Sam Chiu-wai Wong. Tight bounds for approximate carathéodory and beyond. In *International Conference on Machine Learning*, pages 2440–2448, 2017.

[48] Yu Nesterov. Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, 1998.

[49] Donald J Newman. Location of the maximum on unimodal surfaces. *Journal of the ACM (JACM)*, 12(3):395–398, 1965.

[50] Brendan Odonoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.

[51] Harald Räcke, Chintan Shah, and Hanjo Täubig. Computing cut-based hierarchical decompositions in almost linear time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 227–238. Society for Industrial and Applied Mathematics, 2014.

[52] Luis Rademacher. Approximating the centroid is hard. In *Proceedings of the 23rd ACM Symposium on Computational Geometry, Gyeongju, South Korea, June 6-8, 2007*, pages 302–305, 2007.

[53] Sushant Sachdeva, Nisheeth K Vishnoi, et al. Faster algorithms via approximation theory. *Foundations and Trends® in Theoretical Computer Science*, 9(2):125–210, 2014.

[54] Jonah Sherman. Nearly maximum flows in nearly linear time. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 263–269. IEEE, 2013.

[55] Jonah Sherman. Area-convexity, l &infty; regularization, and undirected multicommodity flow. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 452–460. ACM, 2017.

[56] Naum Z Shor. Cut-off method with space extension in convex programming problems. *Cybernetics and systems analysis*, 13(1):94–96, 1977.

[57] Miklós Simonovits. How to compute the volume in high dimension? *Math. Program.*, 97(1-2):337–374, 2003.

[58] Lloyd N Trefethen and JAC Weideman. The exponentially convergent trapezoidal rule. *SIAM Review*, 56(3):385–458, 2014.

[59] Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[60] Pravin M Vaidya. A new algorithm for minimizing convex functions over convex sets. In *Foundations of Computer Science, 1989., 30th Annual Symposium on*, pages 338–343. IEEE, 1989.

[61] S. Vempala. Geometric random walks: A survey. *MSRI Combinatorial and Computational Geometry*, 52:573–612, 2005.

[62] S. S. Vempala. *The Random Projection Method.* AMS, 2004.

[63] Andre Wibisono. Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. *arXiv preprint arXiv:1802.08089*, 2018.

[64] David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

[65] David B Yudin and Arkadii S Nemirovski. Evaluation of the information complexity of mathematical programming problems. *Ekonomika i Matematicheskie Metody*, 12:128–142, 1976.