1 Integration

1.1 Computing the volume of a convex body

In this section, we are interested in designing an effective algorithm with the following inputs and outputs: Given

- a membership oracle for the convex body $K \subseteq \mathbb{R}^n$;
- $x_0 \in K$, $r, R > 0$ such that $x_0 + rB_n \subseteq K \subseteq RB_n$ (where $B_n$ is the unit ball in dimension $n$); and
- $\varepsilon > 0$,

the algorithm gives $V$ such that

$$(1 - \varepsilon)\text{Vol}(K) \leq V \leq (1 + \varepsilon)\text{Vol}(K).$$

Some exact formulas are known to calculate the volume of some simple convex bodies, namely, parallelepipeds, balls and simplices. Therefore, one can think of dividing the convex body $K$ into some of these simple forms.

**Naive algorithm 1 (for polytopes):** Divide the polytope into simplices. *But the number of simplices needed to divide a polytope is exponential in terms of the dimension!*

**Naive algorithm 2:** Partition the convex body $K$ by cubes of edge length $\delta$ (like a pixelisation). *But once again, there is an exponential number of cubes, and we need to call the membership oracle for each of them.*

**Naive algorithm 3:** Find an ellipsoid $E$ such that $E \subseteq K \subseteq n^{3/2}E$ (this can be done in polynomial time). *But it only gives an $n^n$-approximation: $\text{Vol}(E) \leq \text{Vol}(K) \leq (n^{3/2})^n\text{Vol}(E)$.*

Actually, it has been proved that computing the volume is difficult:

**Theorem 1 ([Ele86, BF87]).** Let $a > 0$. For any deterministic algorithm that uses $n^a$ (resp. $2^{2^{n^a}}$) oracle calls, and computes, for all convex body $K \in \mathbb{R}$, $A(K)$ and $B(K)$ such that $A(K) \leq \text{Vol}(K) \leq B(K)$, there exists a convex body $K_0$ such that

$$\frac{B(K_0)}{A(K_0)} \geq \left(\frac{\text{cst} \ n}{a \ln n}\right)^{n/2} \quad \text{(resp.} \quad \geq 2^{\text{cst} \ n}\text{).}$$

**Proof.** Assume that $K = B_n$ is the unit ball in dimension $n$. Assume that the algorithm has already done $m$ oracle calls (and also assume that the $m$ points were in fact inside $K$). For all $i \in \{1, \ldots, m\}$, we thus know that $x_i \in K = B_n$ and thus we know that the ball of diameter $[0, x_i]$, denote by $B^{(i)}$ is included in $B_n$. Moreover, for all $i \in \{1, \ldots, m\}$, we have that $\text{Vol}(B^{(i)}) \leq \frac{\text{Vol}(B_n)}{2^n}$ because the diameter of $B^{(i)}$ is at most half of the diameter of $K$. Therefore, $\text{Vol}\left(\bigcup_{i=1}^m B^{(i)}\right) \leq \frac{\text{Vol}(B)}{2^n}.$
To conclude the proof, it is enough to show that \( \text{conv}(x_1, \ldots, x_m) \subseteq \bigcup_{i=1}^{m} B^{(i)} \). Assume that \( y \in \text{conv}(x_1, \ldots, x_m) \) and \( y \notin \bigcup_{i=1}^{m} B^{(i)} \). This second assumption implies that, for all \( i \in \{1, \ldots, m\} \), the angle \( (0, y, x_i) < \pi/2 \), which is contradiction with \( y \in \text{conv}(x_1, \ldots, x_m) \).

Therefore, after \( m \) oracle calls, we cannot get better than a \( 2^n \) approximation, which concludes the proof.

Bárány and Füredi [BF87] also proved the following equivalent of Theorem 1: for all \( \alpha > 0 \), an algorithm with complexity \( (1/\alpha)^n \) cannot give better than a \( (1 + \alpha)^n \)-approximation. However, the following result holds:

**Theorem 2** ([DV06]). There exists a deterministic algorithm that finds a \( (1 + \alpha)^n \)-approximation with complexity \( (1/\alpha)^O(n) \).

Random sampling then permits to obtain more efficient algorithms:

**Theorem 3** ([DFK91]). For all \( \delta, \alpha > 0 \), there exists a randomised algorithm that computes, with probability \( 1 - \delta \), a \( (1 + \alpha)^n \)-approximation with polynomial complexity in \( (n, \log R, 1/\alpha \log 1/\delta) \).

A naive randomised algorithm could be the following: take a ball \( B \) including the convex body \( K \), sample uniform random points in \( B \) and approximate the volume of \( K \) by the proportion of random points that belong to \( K \) times the volume of the ball \( B \). The problem is that almost all points won’t belong to \( K \).

The [DFK91] algorithm is the following: Assume that \( B_n \subseteq K \subseteq RB_n \). Let \( m = n \log_2 R \), and for all \( i \in \{1, \ldots, m\} \), define

\[
K_i = K \cap (2^{i/n} B_n).
\]

Note that

\[
\text{Vol}(K) = \text{Vol}(B_n) \prod_{i=1}^{m} \frac{\text{Vol}(K_i)}{\text{Vol}(K_{i-1})}.
\]

Then, for \( i \) from 1 to \( m \), sample \( k_i \) points in \( K_{i-1} \) and estimate \( \frac{\text{Vol}(K_i)}{\text{Vol}(K_{i-1})} \) by the proportion of points falling into \( K_i \). Multiply these estimates to get an approximation for \( \text{Vol}(K) \). Note that the estimation of \( \frac{\text{Vol}(K_i)}{\text{Vol}(K_{i-1})} \) by uniform random sampling works because \( \text{Vol}(K_i) \leq 2\text{Vol}(K_{i-1}) \).

To get the complexity, we need to know how many points are needed to get a good approximation at each step. By Tchbychev’s inequality, we know that \( \frac{m^2}{\alpha^2} \) points are enough to get a \( (1 + \alpha) \)-approximation. Therefore, in total, we need \( \frac{m^3}{\alpha^2} = \frac{n^3 \log R}{\alpha^2} \) samples and oracle calls. But in fact,

**Theorem 4** ([DFK91]). \( O\left(\frac{m^2}{\alpha^2}\right) \) samples suffice.
Proof. The idea of the proof is just the following: given $Y_1, \ldots, Y_m$ $m$ i.i.d. random variables, we have that
\[
\frac{\text{Var}(Y_1 \cdots Y_m)}{(E(Y_1 \cdots Y_m))^2} = \prod_{i=1}^{m} \left(1 + \frac{\text{Var}Y_i}{(EY_i)^2}\right) - 1 = \exp \left( m \cdot \frac{\text{Var}Y_1}{(EY_1)^2} \right) - 1.
\]
In our case, $\frac{\text{Var}Y_i}{(EY_i)^2} \sim \frac{\text{const}}{k_i}$ where $k_i$ is the number of points drawn at step $i$. Thus, choosing $k_i = \frac{m}{\alpha^2}$ gives
\[
\frac{\text{Var}(Y_1 \cdots Y_m)}{(E(Y_1 \cdots Y_m))^2} \sim \frac{m^2}{\alpha^2},
\]
which concludes the proof by Tchebychev’s inequality.

This algorithm by Dyer, Frieze, and Kannan has inspired a wide literature, aiming at reducing its complexity. The original algorithm stated above uses the sample algorithm, of complexity $O(n^3)$, and has overall complexity equal to $O(n^{23})$. The table below lists the different improvements of this algorithm and their complexity:

<table>
<thead>
<tr>
<th>authors</th>
<th>complexity</th>
<th>main idea</th>
</tr>
</thead>
<tbody>
<tr>
<td>[DFK91]</td>
<td>$n^{23}$</td>
<td></td>
</tr>
<tr>
<td>[LS90]</td>
<td>$n^{16}$</td>
<td>isoperimetry</td>
</tr>
<tr>
<td>[Lov90]</td>
<td>$n^{10}$</td>
<td>ball-walk</td>
</tr>
<tr>
<td>[DF88]</td>
<td>$n^8$</td>
<td></td>
</tr>
<tr>
<td>[LS92]</td>
<td>$n^7$</td>
<td>rounding + many tools</td>
</tr>
<tr>
<td>[KLS97]</td>
<td>$n^5$</td>
<td>isotropic positions</td>
</tr>
<tr>
<td>[LV06]</td>
<td>$n^4$</td>
<td>hit-and-run</td>
</tr>
<tr>
<td>[CV14]</td>
<td>$n^3$</td>
<td>Gaussian cooling</td>
</tr>
</tbody>
</table>

Surprisingly, a way to get a better complexity is to tackle a more complicated problem: namely the integration on a convex body.

### 1.2 Logconcave Integration

In the previous section, we computed the volume of a convex body $K$ by constructing a sequence of bodies that converge to $K$, computing the volume change for each body. We now shift our focus from volume to integration, which can be viewed as a generalization of volume computation. We begin by formally stating the integration problem.

**Problem 1.** Given as input:

- A membership oracle to a convex body $K \subseteq \mathbb{R}^n$.
- A point $x_0 \in \mathbb{R}^n$ and a number $R \in \mathbb{R}$ such that $x_0 + B_n \subseteq K \subseteq R$.
- An oracle to a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^+$ such that $\int_K f(x) \, dx < \infty$.
- An error parameter $\varepsilon > 0$.

Output a number $V$ such that
\[
(1 - \varepsilon) \int_K f(x) \, dx \leq V \leq (1 + \varepsilon) \int_K f(x) \, dx.
\]

The approach we use for integration is similar to that of volume, where we use a sequence of functions that connect an “easy” function to our target function. For a sequence of functions $\{f_0, \ldots, f_m\}$ where each $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, we rewrite $\int_K f(x) \, dx$ as
\[
\int_K f(x) \, dx = \int_K f_0(x) \, dx \cdot \frac{\int_K f_1(x) \, dx}{\int_K f_0(x) \, dx} \cdot \ldots \cdot \frac{\int_K f_m(x) \, dx}{\int_K f_m(x) \, dx}.
\]
We want \( f_0 \) to be a function which is easy to integrate over \( K \) (perhaps approximately), and then we want to estimate each integral ratio
\[
\frac{\int_K f_i(x) \, dx}{\int_K f_{i-1}(x) \, dx}.
\]

To estimate this ratio, sample a point \( X \) with density proportional to \( f_{i-1} \) and set \( Y = \frac{f_i(X)}{f_{i-1}(X)} \). The expectation of \( Y \) is the quantity we wish to estimate.

**Claim 1.** For \( Y \) and \( f_i \) as defined above,
\[
E(Y) = \frac{\int_K f_i(x) \, dx}{\int_K f_{i-1}(x) \, dx}.
\]

**Proof.** We have that
\[
E(Y) = \int_K \frac{f_i(x)}{f_{i-1}(x)} \cdot \frac{f_{i-1}(y)}{f_{i-1}(y)} \, dy = \frac{\int_K f_i(x) \, dx}{\int_K f_{i-1}(x) \, dx}.
\]

The function \( f_i \) should be “close” to \( f_{i-1} \), so that the ratio of the integrals will be easy to estimate within a target relative error (i.e. the variance \( E(Y^2)/E(Y)^2 \) should be bounded). We now sketch the algorithm.

\begin{verbatim}
Integrate(K, f, ε)
1. Compute (or estimate) \( \int_K f_0 \), call this quantity \( R_0 \).
2. For \( i = 1, \ldots, m \):
   (a) Compute an estimate \( R_i \) of the integral ratio \( \int_K f_i / \int_K f_{i-1} \).
3. Return \( R_0 R_1 \ldots R_m \) as the estimate for \( \int_K f \).
\end{verbatim}

**Figure 2:** General algorithm for integration

We now describe one way to select the sequence of functions \( \{f_0, \ldots, f_m\} \) based on the algorithm in [LV06]. Set \( f_i(x) = e^{-a_i \|x\|} \) and

- \( a_0 = 4n \)
- \( a_i = a_{i-1} \cdot (1 - 1/\sqrt{n}) \) for \( i = 1, \ldots, m - 1 \)
- \( a_m = ε/(2R) \).

The proof of the variance bound will use the following lemma about logconcave functions, whose proof is deferred to the end of the section.

**Lemma 1** ([LV06]). If \( a > 0 \), \( Z(a) = a^n \int_K f(ax) \, dx \), and \( f : \mathbb{R}^n \to \mathbb{R} \) logconcave, then \( Z(a) \) is a logconcave function of \( a \).

**Proof.** (of Lemma 1) Define
\[
G(t, x) = \begin{cases} 
1 & \text{if } t > 0 \text{ and } x \in tK \\
0 & \text{otherwise}
\end{cases}
\]
which is a logconcave function. Also define $F(t,x) = f(x) \cdot G(t,x)$. Since $f,G$ are logconcave, $F$ is also logconcave. Since $F$ is logconcave, its marginal is logconcave. The marginal of $F$ in $t$ is

$$\int_{\mathbb{R}^n} f(x)G(x,t) \, dx = t^n \int_{K} f(tx) \, dx.$$

\[ \square \]

**Lemma 2 ([LV06]).** Let $f_i = e^{-a_i \|x\|}$, $a_i = a_{i-1}(1 - 1/\sqrt{n})$, and $X$ be a random sample with density proportional to $f_{i-1}$. Then, for $Y = f_i(X)/f_{i-1}(X)$, we have that

$$\frac{\mathbb{E}(Y^2)}{\mathbb{E}(Y)^2} \leq 4.$$

**Proof.** For convenience, define $F(a) = \int_K e^{-a \|x\|} \, dx$. From Claim 1, we have that

$$\mathbb{E}(Y) = \frac{F(a_i)}{F(a_{i+1})}.$$

We also derive the second moment:

$$\mathbb{E}(Y^2) = \int_K \left( \frac{f_i(x)}{f_{i-1}(x)} \right)^2 \cdot \frac{f_{i-1}(x)}{\int_K f_{i-1}(y) \, dy} \, dx = \frac{\int_K e^{-2a_i \|x\|}, e^{a_i-1 \|x\|} \, dx}{\int_K f_{i-1}(x) \, dx} F(a_{i-1}) = \frac{F(2a_i - a_{i-1})}{F(a_{i-1})}.$$

We therefore have that

$$\frac{\mathbb{E}(Y^2)}{\mathbb{E}(Y)^2} = \frac{F(2a_i - a_{i-1})F(a_{i-1})}{F(a_i)^2}.$$

Define $Z(a) = a^n F(a)$. By Lemma 1, we have that $Z(a)$ is a logconcave function of $a$. Therefore,

$$\frac{Z(2a_i - a_{i-1})Z(a_{i-1})}{Z(a_i)^2} \leq 1,$$

which after rearranging terms gives

$$\frac{\mathbb{E}(Y^2)}{\mathbb{E}(Y)^2} \leq \left( \frac{a_i^2}{(2a_i - a_{i-1})a_{i-1}} \right)^n = \left( \frac{1}{(2 - (a_{i-1}/a_i))(a_{i-1}/a_i)} \right)^n = \left( \frac{1}{(1 + 1/\sqrt{n})(1 - 1/\sqrt{n})} \right)^n = \left( \frac{1}{1 - 1/n} \right)^n = \left( 1 + \frac{1}{n-1} \right)^n \leq 4.$$

\[ \square \]

We recall well-known properties of logconcave functions.

**Theorem 5.** Marginals of logconcave functions are logconcave. Logconcave functions are closed under convolution.
The following theorem is commonly known as the Prékopa-Leindler inequality.

**Theorem 6.** Suppose \( f, g, h : \mathbb{R}^n \to \mathbb{R}_+ \) are integrable and that \( \forall x, y \in \mathbb{R}^n, \lambda \in [0, 1], h(\lambda x + (1 - \lambda)y) \geq f(x)^\lambda g(y)^{1-\lambda} \). Then

\[
\int_{\mathbb{R}^n} h(x) \, dx \geq \left( \int_{\mathbb{R}^n} f(x) \, dx \right)^\lambda \left( \int_{\mathbb{R}^n} g(x) \, dx \right)^{1-\lambda}.
\]

**Proof.** We prove the lemma by induction on the dimension \( n \). First consider \( n = 1 \). Let \( L_f(t) = \{ x : f(x) \geq t \} \) be a level set of \( f \). Since \( f \) is logconcave, we have that the level sets of \( f \) are convex. Then

\[
\lambda L_f(t) + (1 - \lambda)L_g(t) = \{ \lambda x + (1 - \lambda)y : f(x) \geq t, g(x) \geq t \} \subseteq L_h(t)
\]

since \( h(\lambda x + (1 - \lambda)y) \geq t \). Therefore, we have that \( \text{vol}(L_h(t)) \geq \lambda L_f(t) + (1 - \lambda)L_g(t) \) for all \( \lambda \in [0, 1] \) and

\[
\int_{\mathbb{R}} h(x) \, dx = \int_0^\infty \text{vol}(L_h(t)) \, dt \\
\geq \lambda \int_0^\infty L_f(t) \, dt + (1 - \lambda) \int_0^\infty L_g(t) \, dt \\
\geq \left( \int_{\mathbb{R}} f(x) \, dx \right)^\lambda \left( \int_{\mathbb{R}} g(x) \, dx \right)^{1-\lambda}.
\]

Now suppose the inequality is true for dimension \( n - 1 \). Define \( h(z, x) = h_z(x) \) for \( z \in \mathbb{R}, x \in \mathbb{R}^{n-1} \). (similarly for \( f, g \)). Fix a \( z \). Then the marginal distribution on the remaining \( n - 1 \) coordinates is logconcave. Then for \( z = \lambda z_1 + (1 - \lambda)z_2 \), by a similar argument to \( n = 1 \)

\[
h(\lambda z_1 + (1 - \lambda)z_2, \lambda x_1 + (1 - \lambda)x_2) \geq f(z_1, x_1)^\lambda g(z_2, x_2)^{1-\lambda},
\]

which implies that

\[
h_z(\lambda x_1 + (1 - \lambda)x_2) \geq f_z(1)^\lambda g_z(2)^{1-\lambda}.
\]

By induction, we have that

\[
\int_{\mathbb{R}^{n-1}} h_z(x) \, dx \geq \left( \int_{\mathbb{R}^{n-1}} f_z(x) \, dx \right)^\lambda \left( \int_{\mathbb{R}^{n-1}} g_z(x) \, dx \right)^{1-\lambda},
\]

and thus

\[
\int_{\mathbb{R}^n} h(x) \, dx \geq \left( \int_{\mathbb{R}^n} f(x) \, dx \right)^\lambda \left( \int_{\mathbb{R}^n} g(x) \, dx \right)^{1-\lambda}.
\]

We now give a slightly more detailed algorithm for integration, which will work for any logconcave function.

| Integrate\((K, f, \varepsilon)\) |
|---------------------------------
| 1. Set \( f_i(x) = f(x)^{a_i}, x \in K \). |
| 2. Set \( a_0 = 0, a_m = 1 \), and \( a_{i+1} = a_i(1 - 1/\sqrt{i}) \) for \( i = 0, \ldots, m - 2 \). |
| 3. For \( i = 1, \ldots, m \), compute \( w_i = \int f_i / \int f_{i-1} \). |
| 4. Output \( W_1 \ldots W_m \cdot \int f_0 \). |

We note that for optimizing a logconcave function \( f \), we can use a slightly different cooling schedule and instead of estimating integral ratios, we simply output the point \( x \) with the largest function value \( f(x) \) that we see. So, integrating and optimizing a general logconcave function are very closely related.
References


