Chapter 1

Newton-Okounkov Bodies

1.0.1 Theorem (Bernstein-Kushnirenko) *Let $f_1 \cdots f_n = 0$ be Laurent polynomials in n variables. Read off the exponents of the monomials that show up for each f. This ignores the coefficients. Call these sets $A_i \subset \mathbb{Z}^n$ for each f_i. Let Δ_i be the convex hull thereof. Th number of solutions when the coefficients are generic is $n!V(\Delta_1 \cdots \Delta_n)$ is the Minkowski mixed volume. At an extreme, letting A_i all be equal then gives $n!V(\Delta)$ with usual Euclidean volume.*

1.0.2 Definition (Mixed Volume)

$$V(\Delta_1 \cdots \Delta_n) = \frac{1}{n!} \frac{d^n}{d\lambda_1 \cdots d\lambda_n} Vol(\lambda_1 \Delta_1 + \cdots \lambda_n \Delta_n) \mid_{\lambda_1 \cdots \lambda_n \to +0}$$

where the $+$ inside Vol indicates Minkowski sum.

Chapter 2

Knapsack Polytope

2.1 Knapsack Problem

2.1.1 Definition (Knapsack Problem) *Let $A_1 \cdots A_n$ be goods with values v_i and weights w_i. The knapsack can include any natural number of each good with the condition that the total weight should be less than W. Call the numbers of each as x_i.*

$$\begin{aligned} V &= \sum v_i x_i \\ \sum w_i x_i &\leq W \end{aligned}$$

The goal is to maximize V.

2.1.2 Definition (Decision Problem Formulation) *Given a V_0, can we find a solution with $V \geq V_0$?*

2.1.3 Lemma (NP-Complete) *This decision problem is NP-complete. The optimization problem is NP-hard.*

2.1.4 Lemma (Augmented Form) *The weight inequality can be rephrased by introducing another natural number s to represent $W - \sum w_i x_i$. The system then becomes solving the following system in $\mathbb{N}^{n+2}$.*

$$\begin{pmatrix} 1 & -v_1 & \cdots & -v_n & 0 \\ 0 & w_1 & \cdots & w_n & 1 \end{pmatrix} \begin{pmatrix} V \\ x_1 \\ \cdots \\ x_n \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ W \end{pmatrix}$$

Another way is given by

$$\begin{pmatrix} v_1 & \cdots & v_n & 0 \\ w_1 & \cdots & w_n & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \cdots \\ x_n \\ s \end{pmatrix} = \begin{pmatrix} V \\ W \end{pmatrix}$$

The solution with largest V is sought.

2.1.5 Lemma (0-1 Algorithm) *Let $m[i, j]$ be the maximum value attained by using items $\leq i$ and weight $\leq j$. In particular $i = 0$ is the case when no items are allowed to be used and so no value can be attained no matter the weight.*

Listing 2.1: KnapsackDynamicProgramming

```
for j from 0 to W do:
    m[0, j] := 0

for i from 1 to n do:
    for j from 0 to W do:
        if w[i] > j then:
            m[i, j] := m[i-1, j] \% We can't use item i with weight restriction j
        else:
            m[i, j] := max(m[i-1, j], m[i-1, j-w[i]] + v[i]) \% We can use it so
```

2.2 Ehrhart Polynomial

2.2.1 Definition (Ehrhart Polynomial) *For a polytope $\mathcal{P} \subset \mathbb{R}^d$ with integral vertices, the count of points in $t\mathcal{P} \bigcap \mathbb{Z}^d$ as a function of positive integer t denoted $L_{\mathcal{P}}(t)$.*

2.2.2 Theorem (Ehrhart) *$L_{\mathcal{P}}(t)$ is a polynomial of degree d. The leading term is the relative volume. $L_{\mathcal{P}}(0) = \chi(\mathcal{P})$ and $L_{\mathcal{P}}(-t) = (-1)^d L_{\mathcal{P}^o}(t)$ where χ is Euler characteristic and $\mathcal{P}^o$ denotes relative interior.*

If the vertices are rational but not integral, we get a quasipolynomial instead.

2.2.3 Definition (Quasipolynomial) *An expression of the form $c_d(t)t^d + \cdots c_1(t)t + c_0(t)$ with the $c_i(t)$ periodic functions in t.*

2.3 Ehrhart Polynomial of Linear Programs

2.3.1 Theorem *Let $\mathcal{P}$ be the polytope given by $x \in \mathbb{R}^d_{\geq 0}$ and constraint $Ax = b$ where $A \in M_{m \times d}(\mathbb{Z})$ and $b \in \mathbb{Z}^m$ are integral. $\mathcal{P}$ has rational vertices, not necessarily integral. Then the Ehrhart quasipolynomial is given by*

$$L_{\mathcal{P}}(t) \quad = \quad \frac{1}{(2\pi i)^m} \int_{|z_1|=\epsilon_1} \cdots \int_{|z_m|=\epsilon_m} \frac{z_1^{-tb_1-1} \cdots z_m^{-tb_m-1}}{(1-\mathbf{z}^{\mathbf{c_1}}) \cdots (1-\mathbf{z}^{\mathbf{c_d}})}$$

where $\mathbf{z}^{c_j}$ stands for the product $z_1^{A_{1j}} \cdots z_m^{A_{mj}}$. That is $\mathbf{z}$ stands for z_1 through z_m and $\mathbf{c_j}$ is the j'th column of A. Also $0 < \epsilon_1 \cdots \epsilon_m < 1$ are distinct real numbers.

Proof https://arxiv.org/pdf/math/0202267.pdf ☐

Looking at the second linear program phrasing of the knapsack problem givs $m = 2$ and $d = n+1$ with $V = b_1$. $L_{\mathcal{P}}(1)$ counts the number of solutions and we wish to see how it depends on b_1. For some critical value, V_{crit}, the point count will be 0 from then on. That means the optimal value that can be attained will be $V_{crit} - 1$. That is the last case when the polytope has integral points.

$$L_{\mathcal{P}}(t) \quad = \quad \frac{-1}{4\pi^2} \int_{|z_1|=\epsilon_1} \int_{|z_2|=\epsilon_2} \frac{z_1^{-tV-1} z_2^{-tW-1}}{(1 - z_1^{v_1} z_2^{w_1}) \cdots (1 - z_1^{v_n} z_2^{w_n})(1 - z_2)}$$

2.3.2 Corollary *Expanding the term corresponding to the n'th good in a geometric series gives an expansion of $L_{\mathcal{P}}(t)$ for the first $n-1$ goods with total value $V - kv_n$ and total value $W - kw_n$ summed over k. This is obviously interpreted as deciding how many of good n to pack and packing the remaining knapsack with the other $n-1$ goods.*

2.3.3 Lemma *The other values of t get interpreted as what happens if you are allowed to break each item into $\frac{1}{t}$ parts and then pack those into the knapsack. For large t, this tends to the continuous linear programming problem where one is allowed to use arbitrary fractions of the goods.*

2.3.4 Example