1 Shortest Vector Problem: History and Motivation

Recall that for a given basis $B = (b_1, ..., b_d)$, the lattice associated with this basis is defined as

$$\Lambda = \{\sum_{i=1}^{d} \lambda_i b_i | \lambda_i \in \mathbb{Z}\}$$

Then, continuing from last lecture, we want to solve the problem of knowing if an integer point exists in a convex body. We do this in three steps:

1. "Round" this convex body
2. Find a "good" basis
3. Either show that there exists a point in the convex body, or enumerate over lower dimensional problems

In the previous lecture, we discussed using the Gauss-Lagrange approach to find such a basis for the 2D case. Today's lecture will focus on finding a "good" basis in the general case. We loosely define "good" as being close to orthogonal (by some notion of close), and having short vectors. Intuitively, this is because having short vectors enables us to take small steps when stepping through the lattice, making it easier to find the point(s) inside the convex body. This, then, motivates the following problem:

Shortest Vector Problem: Given a lattice Λ, find the shortest non-zero vector. The length of the solution to this problem is denoted SVP(Λ).

Recall Minkowski's theorem from the previous lecture, which states that $\exists$ a nonzero vector in any lattice of length $\sqrt{d} \cdot (\det(\Lambda))^{1/d}$. The goal of today's lecture will be to introduce the **LLL algorithm**, and prove the following theorem:

Theorem 12.1. *The LLL algorithm returns a vector of length* $\leq 2^{\frac{d-1}{2}} SVP(\Lambda)$.

1.1 History

Consider the following problem: Given $\alpha_1, ..., \alpha_d$, find $P_1, ..., P_d \in \mathbb{Z}$, $q \in \mathbb{Z}$, $q \leq Q$ (for some fixed Q), s.t.

$$|\alpha_i - \frac{p_i}{q}| \leq \text{small}$$

A simple initial solution to this would be to set $q = Q$, $P_i = [\alpha_i Q]$ (the closest integer to $\alpha_i Q$). This gives us an error $\leq \frac{1}{2q}$, since P_i is a factor of $\frac{1}{Q} = \frac{1}{q}$ away from the closest integer. Can we do better? That is, could we get $\frac{\epsilon}{q}$ for any $\epsilon > 0$? The following theorem states that this is indeed possible.

Theorem 12.2. *There exists a solution to the above problem with error* $\frac{1}{qQ^{1/d}}$. *Then, for any* $\epsilon > 0$, *we can choose a big enough* Q *to get* $\epsilon = \frac{1}{Q^{1/d}}$.

Proof. We can rewrite this problem as follows: find integers p_i, q s.t.

$$\forall i, |q\alpha_i - p_i| \leq \epsilon, |q| \leq Q$$

In other words, consider the body created by the range ϵ away from the line $q\alpha_i$ from $-Q$ to Q in every coordinate (as shown in Figure 12.1). The volume of this body is clearly

$$(2Q)(2\epsilon)^d = 2^{d+1}Q\epsilon^d = 2^{d+1}$$

since we chose $\epsilon = \frac{1}{Q^{1/d}}$. By Minkowski's theorem, $\exists p_1, ..., p_d$ inside this body, as desired.

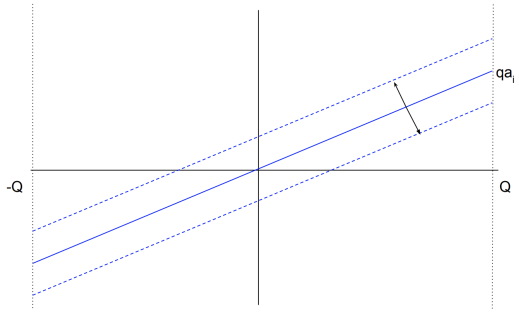

Figure 12.1: Fixed coordinate diagram of convex body

□

This theorem gives us the existence of such vectors. We want to be able to find the vectors themselves.

2 LLL Algorithm

Recall from the previous lecture the Gauss-Lagrange algorithm, in which we subtracted the integer part of $\mu = \frac{\langle b_2, b_1 \rangle}{\|b_1\|^2}$ copies of b_1 from b_2, using Gram-Schmidt as motivation (remember, we want these vectors to be close to orthogonal). Now, we want to extend this to the case with dimension d. As in the Gauss-Lagrange algorithm, we want $\mu \leq \frac{1}{2}$ (for each pair). Furthermore, we want to ensure the lengths don't fall too fast (if they decrease, they don't do so by much). This intuition is formalized in the following theorem:

Theorem 12.3. *We want to maintain an ordering of* $b_1, ..., b_d \to b_1^*, ..., b_d^*$, *where for* $\mu_{ij} = \frac{\langle b_j, b_i^* \rangle}{\|b_i^*\|^2}$

$$b_j^* = b_j - \sum_{i<j} \mu_{ij} b_i^*$$

with the following two conditions:

1. *Coefficient reducedness:* $|\mu_{ij}| \leq \frac{1}{2}\ \forall i < j$
2. *Lovasz Condition:* $|b_i^*|^2 < 2|b_{i+1}^*|^2\ \forall i$

Then, the LLL algorithm achieves both conditions.

Proof. We start by introducing the LLL algorithm. This runs in two steps:

1. For some i, j pairs, we subtract copies of b_i from b_j until we have satisfied coefficient reducednes
2. If $\exists i, i+1$ violating the Lovasz condition, we swap i and $i+1$ and go back to step 1. Else, we return b_1.

We run step 1 of the algorithm as follows:

Algorithm 1 Step 1 of LLL

```
for j = 1, ..., d do
    for l = j, ..., 1 do
        μ_lj ← ⟨b_j, b_l^*⟩ / |b_l^*|^2
        if |μ_lj| > 1/2 then
            b_j ← b_j − [μ]b_l
        end if
    end for
end for
```

We want to show that following this algorithm, once we've made $|\mu_{lj}| < \frac{1}{2}$, within this round, it will stay that way. Suppose we have $b_1, ..., b_l, ..., b_j, ..., b_n$ and $|\mu_{lj}| > \frac{1}{2}$. Then, we want to set $b_j = b_j - c \cdot b_l$, for some integer c. We want to show that μ_{ik} remains unchanged for any i, k pair that has already been visited. Recall that $\mu_{ik} = \frac{\langle b_k, b_i^* \rangle}{|b_i^*|^2}$. We consider two cases, as demonstrated in the matrix in Figure 12.1: 1. $k < j$. Clearly, b_k does not change as a result of changing b_j.

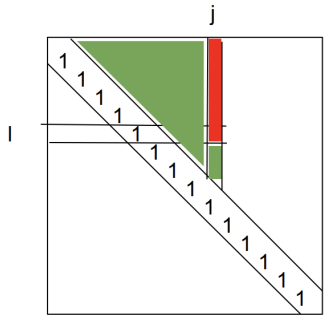

Figure 12.2: Changed and unchanged μ values in matrix

Furthermore, since $i < k < j$, b_i^* only relies on $b_1^*, ..., b_{i-1}^*$ and b_i. Since none of these are changed, this μ term is also unchanged, as desired.
2. $k = j, i > l$. Here, notice that b_l must be orthogonal to b_i^*. This is because b_j can be written as a linear combination of $b_1^*, ..., b_j^*$ by construction, and b_i^* is constructed by subtracting the span of $b_1^*, ..., b_{i-1}^*$, which is equivalent to the span of $b_1, ..., b_{i-1}$ from b_i. Therefore, the dot product of $\langle b_j, b_i^* \rangle$ will remain unchanged despite subtracting b_l from b_j.

The last thing for us to show in this algorithm is that this algorithm terminates in polynomial time.

Lemma 12.4. *Suppose a_1, a_2 are vectors s.t. $|\mu_{12}| \leq \frac{1}{2}$ and $|a_2^*|^2 \leq \frac{1}{2}|a_1^*|^2$. Then,*

$$|a_2|^2 \leq \frac{3}{4}|a_1^*|^2 = \frac{3}{4}|a_1|^2$$

.

Proof.

$$a_2 = a_2^* + \mu_{12}a_1 = a_2^* + \mu_{12}a_1^*$$

Since a_2^* and a_1^* are orthogonal, we get

$$|a_2|^2 = |a_2^*|^2 + \mu_{12}^2|a_1^*|^2$$

Using the two assumptions, we get $|a_2^*|^2 \leq \frac{1}{2}|a_1^*|^2$, and $\mu_{12}^2|a_1^*|^2 \leq \frac{1}{4}|a_1^*|^2$. Therefore,

$$|a_2|^2 \leq \frac{3}{4}|a_1^*|^2 = \frac{3}{4}|a_1|^2$$

as desired. □

This tells us that if the Lovasz condition was violated, i.e. $|a_2^*|^2 \leq \frac{1}{2}|a_1^*|$, then that must mean $|a_2|$ was smaller than $|a_1|$ by a factor of $\sqrt{3/4}$. Therefore, when we swap the two, the new a_1 will be smaller than the old one by a factor of $\sqrt{3/4}$. We can extend the same reasoning to show that if we condition on $b_1^*, ..., b_{i-1}^*$ and swap b_i and b_{i+1} (because the Lovasz condition was violated), then the new $|b_i|$ is smaller than the old one by a factor of $\sqrt{3/4}$.
Consider the potential function

$$\Phi = \prod_{i=1}^{d} \text{vol}_i(b_1, ..., b_i)$$

We leave it as an exercise to verify that this is equal to

$$\prod_{i=1}^{d}(\prod_{j\leq i}|b_j^*|) = \prod_{i=1}^{d}|b_i^*|^{d-i+1}$$

Notice, then, that if we swap i and $i+1$, all the volumes stay the same except vol_i, which is reduced by $\sqrt{3/4}$. This algorithm only repeats if it performs a swap, and everytime it does, the potential decreases by at least $\sqrt{3/4}$. The initial potential is at most $(\text{poly}(d)\max_{ij}|B_{ij}|)^{\text{poly}(d)}$. Since we reduce this potential by a constant factor at every step, the overall algorithm runs for at most $\text{poly}(d, \log\max_{ij} B_{ij})$ steps. Therefore, the LLL algorithm is efficient and achieves both conditions, as desired.

□

Lastly, we show that this algorithm correctly outputs a $2^{\frac{d-1}{2}}$ approximation to SVP.

Theorem 12.5. *b_1 is a $2^{\frac{d-1}{2}}$ approximation to SVP*

Proof. By the end of the algorithm, we know that for all i,

$$|b_1|^2 = |b_1^*|^2 \leq 2^{i-1}|b_i^*|^2$$

(this follows inductively from the Lovasz condition). Since $i \leq d$, we have that for all i

$$|b_1|^2 \leq 2^{d-1}|b_i^*|^2$$

Since this is true for all i, it must be true for the minimum. Therefore,

$$|b_1|^2 \leq 2^{d-1} \min_i |b_i^*|^2$$

$$|b_1| \leq 2^{\frac{d-1}{2}} \min_i |b_i^*|$$

Now, we want to show that $v \geq \min_i |b_i^*|$. Note that if this is true, then b_1 is a $2^{\frac{d-1}{2}}$ approximation to SVP, and we are done.
Consider the shortest vector v in this lattice. $v = \lambda_1 b_1 + ... + \lambda_l b_l$, where λ_l is the last nonzero coefficient. If $l = 1$, then $|v| \geq |b_1|$, and we are done. Else, let H be the span of $b_1, ..., b_{l-1}$. v cannot be a vector along the plane H, or λ_l would be 0. However, the distance to the next plane is, by definition, b_l^*, as demonstrated in Figure 12.3:

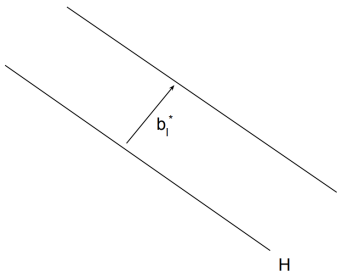

Figure 12.3: H plane and b_l^* vector

Therefore $|v| \geq |b_l^*|$, which implies that $|v| \geq \min_i |b_i^*|$, as desired. □