6.885 Algebra and Computation October 19, 2005

Lecture 11

Lecturer: Madhu Sudan *Scribe: Kyomin Jung*

Today, we will give a definition of lattice in $\mathbb{R}^n$ and study its two specifications, by primal basis and dual basis. Then we will focus on the problem of finding shortest vectors in a given lattice. For this problem, we will present the Gauss algorithm that finds exact solution for the 2 dimensional case, and LLL(Lenstra, Lenstra, Lovasz,'82) algorithm that finds a 2^n approximation solution in polynomial time of n for the n dimensional case.

1 Motivation

Think of the following problem. Given $G(x) \in \mathbb{Z}[x]$, $d, N \in \mathbb{N}$ and $c > 0$,
Find $g = \sum_{i=0}^{d} g_i x^i \in \mathbb{Z}[x], H(x) \in \mathbb{Z}[x]$ such that

$$g(x) = G(x) \cdot H(x) \pmod{\text{N}}$$

where (mod N) means modulo for each coefficients, and $|g_i| \leq c$ for each $0 \leq i \leq d$.

This can be expressed as a system of linear equations,

$$\forall\ 0 \leq i \leq d, \quad g_i = \sum_j G_j H_{i-j} + q_i N$$

where $q_i \in \mathbb{Z}$. We can obtain non-zero integer solutions for g_i's, H_i's and q_i's by solving above linear equations.

But, how can we obtain a solution that satisfies $|g_i| \leq c$?

Naturally this problem induces the concept of lattice and the problem of finding a solution with small non-zero norm.

2 Lattice and its basis

First we give a definition of a lattice in $\mathbb{R}^n$.

Definition 1 *$L \in \mathbb{R}^n$ is a lattice if it satisfies*
1) $x, y \in L \ \rightarrow\ x + y \in L$ and $-x \in L$.
2) L is a discrete set in $\mathbb{R}^n$.

Where a set $A \in \mathbb{R}^n$ is called a discrete set if there exist $\delta > 0$ s.t. $\forall x \in A, \quad Ball(x, \delta) \cap A = \{x\}$.

Definition 2 *For a given finite set $M \in \mathbb{R}^n$, we define*

$$L_M = \{\sum_i c_i m_i | c_i \in \mathbb{Z}, m_i \in M\}.$$

If a lattice L is equal to L_M for some M, we say that M is a (primal) basis of L, and L is generated by M.

Note that for any $M \in \mathbb{R}^n$, L_M is closed under addition and subtraction, but it is not always discrete. A sufficient condition for L_M to be discrete, hence L_M to be a lattice, is, M is finite and $M \in \mathbb{Q}^n$. Now we will show that essentially this is a necessary condition too.

First, we show that any lattice L in $\mathbb{R}^n$ have a finite basis. Let $k = \text{rank(L)}$ as a subset of $\mathbb{R}^n$. Let $\{b_1, b_2, \ldots b_k\} \subset L$ be a linearly independent subset of L. Then there are finitely many points of L that is in the k-dimensional parallelogram formed by $b_1, b_2, \ldots b_k$, because L is discrete. These points together with b_i's form a finite basis of L.

By similar argument, one can show that if L is a lattice in $\mathbb{R}^n$, there exist an invertible linear transform $T : \mathbb{R}^n \to \mathbb{R}^n$ so that T sends L into $\mathbb{Q}^n$, and T is "almost" distance preserving. So from now on we will assume that $L \in \mathbb{Q}^n$ and it has a finite basis.

Now we show that given $L \in \mathbb{Q}^n$ with a finite basis $\{b_1, b_2, \dots b_m\}$, we can find a basis of L whose size is equal to the rank of L in time polynomial of n. We may assume that $b_i \in \mathbb{Z}^n$ by multiplying lcm of denominators of b_i. Note that replacing b_i with $b_i + \sum_{j=1, j\neq i}^{m} c_j b_j$, $c_j \in \mathbb{Z}$ gives another basis for L.

So, given a basis

$$\begin{pmatrix} | & | & \dots & | \\ b_1 & b_2 & \dots & b_m \\ | & | & \dots & | \end{pmatrix},$$

we can obtain a new basis of the form

$$\begin{pmatrix} g_1 & 0 & \dots & 0 \\ | & | & \dots & | \\ b_1^{(1)} & b_2^{(1)} & \dots & b_m^{(1)} \\ | & | & \dots & | \end{pmatrix}$$

where g_1 is the gcd of $b'_{1i}s$. Similarly, we can obtain a basis of the form

$$\begin{pmatrix} g_1 & 0 & 0 & \dots & 0 \\ | & g_2 & 0 & \dots & 0 \\ | & | & g_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ | & | & | & \dots g_j \dots & 0 \\ b_1^{(j)} & b_2^{(j)} & b_3^{(j)} & \dots b_j^{(j)} \dots & b_m^{(j)} \\ | & | & | & \dots | \dots & | \end{pmatrix}$$

where g_j is the gcd of $b_{ji}^{(j-1)}$'s, $j \leq i \leq m$. In this way we can find a basis whose size is $k = \text{rank(L)} \leq \text{n}$ in poly(n) time.

3 Dual specification of a lattice

Given $M^* = \{b_1^*, b_2^*, \dots b_k^*\} \in \mathbb{Q}^n$, define

$$L_{M^*}^* = \{v \in Q^n \mid \forall j \in \{1, 2, \dots k\}, < v, b_j^* > \in Z\}.$$

If a lattice $L \in \mathbb{Q}^n$ is equal to $L_{M^*}^*$, we say that M^* is a dual basis of L. Note that if $\text{rank}(M^*) < n$, then $L_{M^*}^*$ cannot be a discrete set because the orthogonal space of $\text{span}(M^*)$ has positive dimension. So a dual basis of a lattice $L \in \mathbb{Q}^n$ has rank n. On the other hand, the set $L_{M^*}^*$ always have rank n because, for any line line l in $\mathbb{Q}^n$ passing through the origin, $l \cap L_{M^*}^*$ is non-empty. So a lattice $L \in \mathbb{Q}^n$ has a dual basis only when L has rank n. Soon we will see that L has a dual basis if L has rank n.

Now if $M^* = \{b_1^*, b_2^*, \dots b_k^*\}$ is a dual basis of L, replacing b_i^* with $b_i^* + \sum_{j=1, j\neq i}^{n} c_j b_j^*, c_j \in \mathbb{Z}$ is also a dual basis of L. So by similar way to the previous primal basis case, we can obtain a dual basis of L having size n.

Now we think of how to find a dual basis of L given its primal basis. Assume that $rank(L) = n$. Let

$$B = \begin{pmatrix} | & | & \dots & | \\ b_1 & b_2 & \dots & b_n \\ | & | & \dots & | \end{pmatrix},$$

be a primal basis of L. Then

$$B^* = \begin{pmatrix} - & b_1^* & - \\ - & b_2^* & - \\ \cdots & \cdots & \cdots \\ - & b_n^* & - \end{pmatrix},$$

is a dual basis of L if and only if $B^*B \in Z^{n\times n}$ and $\det(B^*B) = \pm 1$. So we can obtain $B^* = B^{-1}$ as a dual basis of L. Note that if M^* is a dual basis of L_M, then M is a dual basis of L_{M*}. Usually, given a lattice L, one may specify L by its primal basis or by its dual basis, which is simpler.

4 Shortest Vector problem

For the problem introduced in the beginning of this lecture, the set of all the solutions $g(x) \in \mathbb{Z}[x]$ that satisfies

$$\forall\ 0 \le i \le d,\quad g_i = \sum_j G_j H_{i-j} + q_i N$$

for some $H(x) \in \mathbb{Z}[x], q_i \in \mathbb{Z}$, forms a lattice in $\mathbb{Z}^{d+1}$. So the problem is to find a non zero vector in a given lattice that has short distance from the origin. (In this case, with respect to the l_∞ norm.) Generally, finding a shortest non-zero vector of a given lattice $L \in \mathbb{R}^n$ with respect to the given norm is called "Shortest Vector Problem(SVP)". In 1982, Lenstra, Lenstra, Lovasz gave a $2^{n/2}$ factor approximation polynomial time algorithm for SVP. In this lecture, we will present this algorithm and show that it is a 2^n factor approximation algorithm. As a known result for the hardness of the problem, Ajtai('98) showed that it is NP hard to solve SVP exactly. In 2001, Micciancio showed that SVP is hard to approximate within any constant factor less than $\sqrt{2}$. And in 2004, Khot showed that SVP is hard to approximate within any constant factor (under the assumption that NP≠RP).

5 Gauss Algorithm

In 1801, Gauss gave an algorithm that finds a shortest vector for the case when $n = 2$. Given a lattice $L_{\{v_1,v_2\}} \in \mathbb{R}^2$,

Gauss Algorithm(v_1, v_2):

1. If we can reduce $|v_2|$ by subtracting an integer multiple of v_1 from v_2, do so.
2. If $|v_2| < |v_1|$, swap(v_1, v_2) and goto step 1. Else, output v_1.

This algorithm finds a shortest vector in a polynomial time in the size of input bits.

6 LLL Algorithm

SVP problem in $O(1)$ dimension can be solved in polynomial time (over the number of input bit size) by solving integer programming with $O(1)$ variables. When the dimension n is not bounded, LLL algorithm gives a 2^n approximation in polynomial time over n.

Given a lattice L with a basis $\{b_1, \ldots, b_n\}$,

LLL Algorithm$(b_1, \ldots, b_n)$:

1. For every pair (i, j) with $i < j$, if we can reduce $|b_j|$ by subtracting integer multiple of b_i from b_j, do so. (Until no such a pair exists.)

2. If $|b_i| < \frac{3}{4}|b_{i-1}|$, swap$(b_{i-1}, b_i)$ and goto step 1.

3. If no such i exists, output b_1.

Step 1 of the algorithm can be done in polynomial time by using a similar way of the Gram-Schmidt orthogonalization that gives an orthogonal basis (of the vector space $\mathbb{R}^n$) from a given basis.

Gram-Schmidt orthogonalization$(b_1, \ldots, b_n)$:

- From $i = 1$ to n, $\widetilde{b_i} = b_i - \text{projdection}(\text{b}_\text{i}, \text{span}(\widetilde{\text{b}_1}, \ldots, \widetilde{\text{b}_{\text{i}-1}}))$.
- Output $\{\widetilde{b_1}, \ldots, \widetilde{b_n}\}$.

Similar to the Gram-Schmidt orthogonalization, in step 1 of the LLL algorithm, increase j from 2 to n, and for each fixed j, increase i from 1 to $j-1$. And subtract b_j by $c_{ij}b_i$ where c_{ij} is the nearest integer to $\frac{<b_i, b_j>}{|b_i|^2}$. So it can be done in polynomial time.

The $\frac{3}{4}$ factor in step 2 guarantees that LLL algorithm goes back to step 1 at most polynomially many times over n, so the algorithm runs in polynomial time over n. In the next lecture we will prove that its output gives a 2^n factor approximation to a shortest vector.

7 References

1. A. K. Lenstra, H. W. Lenstra, Jr. and L. Lovasz, Factoring Polynomials with Rational Coefficients, Math. Ann. 261, 1982.

2. Subhash Khot, Hardness of approximating the shortest vector problem in lattices, in Proc. 45th Symposium on Foundations of Computer Science, 2004.

3. D. Micciancio, The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant, in Proc. 39th Symposium on Foundations of Computer Science 1998.

4. M. Ajtai, The shortest vector problem in l 2 is NP-hard for randomized reductions, In Proceedings of the thirtieth annual ACM symposium on theory of computing, 1998.