Algebra and Computation Course Instructor: V. Arvind

Lecture 18 and 19: LLL and Factorization over $\mathbb{Q}$

Lecturer: V. Arvind *Scribe: Ramprasad Saptharishi*

1 Overview

Another problem very essential for factoring univariate polynomials over $\mathbb{Q}$ is the shortest vector problem. Of course, finding the optimum solution is NP-hard and we only want an approximation algorithm to this.

We shall discuss the LLL algorithm for the shortest vector and then give the algorithm for factorizing univariate polynomials over $\mathbb{Q}$.

2 The Shortest Vector Problem

We are given a basis $\{b_i\}_{0\leq i\leq n}$ in R^n and we want to find a vector $v = \sum a_i b_i$, where $a_i \in \mathbb{Z}$, whose norm (the usual euclidian norm) is minimum.

Solving this problem in full generality is NP-hard and we do not expect to find the optimal solution. LLL however allows us to find an approximate solution, the approximation factor depending only on the dimension.

The basic idea is in mimicking the Gram-Schmidt orthogonalization method on a lattice.

2.1 The Gram-Schmidt Orthogonalization

We are given a basis $\{b_1, b_2, \cdots, b_n\}$ and we want to convert it into a new orthogonal basis $\{b_1^\star, b_2^\star, \cdots, b_n^\star\}$.

The GS algorithm is as follows:

$$\begin{aligned} & b_1^\star &=& \ b_1 \\ \forall 1 < i \leq n \qquad & b_i^\star &=& \ b_i - \sum_{j<i} \mu_{ij} b_j^\star \qquad \text{where } \mu_{ij} = \frac{\left\langle b_i, b_j^\star \right\rangle}{\left\| b_j^\star \right\|^2} \end{aligned}$$

The GS basis satisfies the following properties, which are easy to check:

- Different ordering of the basis vectors could give different GS orthogonal bases.

- The basis vectors are mutually orthogonal.
- For each i, $\|b_i^\star\| \leq \|b_i\|$.
- Fore each i, the span of $\{b_1, \cdots, b_i\}$ is the same as the span $\{b_1^\star, \cdots, b_i^\star\}$.
- If B is the matrix whose columns are the vectors $\{b_i\}$ and if $B^\star$ is the matrix with columns $\{b_i^\star\}$, then the tranformation is given by the following unimodular triangular matrix:

$$\begin{bmatrix} 1 & & & \\ \mu_{21} & 1 & & \\ \vdots & & \ddots & \\ \mu_{n1} & \mu_{n2} & \cdots & 1 \end{bmatrix} B^\star = B$$

  and therefore $\det B = \det B^\star$, the volume of the fundamental parallelopiped of the lattice is preserved.
- If $L(B)$ is the lattice generated by B,

$$|\det L(B)| := |\det B| = \|b_1^\star\| \, \|b_2^\star\| \cdots \|b_n^\star\| \leq \|b_1\| \, \|b_2\| \cdots \|b_n\|$$

  And if $\mathcal{B}$ is the largest value in the matrix B, then we have the famous hadamard inequality

$$|\det B| \leq n^{n/2} \mathcal{B}^n$$

2.2 Reduced Basis

The following observation is the key to LLL.

Observation 1. *Let $b = \lambda_i b_i$ be the shortest vector in the lattice. Then*

$$\|b\| \geq min \, \|b_i^\star\|$$

Proof. Let k be the largest index such that $\lambda_k \neq 0$. Since the GS tranformation matrix has 1s on the diagonal, even after we write $b = \sum \lambda_i' b_i^\star$, $\lambda_k = \lambda' k$.

Hence,

$$\|b\| = \sum |\lambda_i'|^2 \, \|b_i^\star\| \geq |\lambda_k| \, \|b_k^\star\| \geq \|b_k^\star\| \geq min \, \|b_i^\star\|$$

□

With this as the motivation, we have the following concept of a reduced basis.

Definition 2. *A basis* $\{b_i\}$ *is said to be a reduced basis if it satisfies the condition that for all* i, $\|b_i\|^2 \leq 2\|b_{i+1}\|^2$.

From the earlier observation, it is clear that once we have a reduced basis,

$$\|b_1\| \leq 2^{\frac{n-1}{2}} \|\text{opt}\|$$

And hence, if we can find a reduced basis for the lattice, then we have achieved our goal of finding a constant factor (only a function of degree) approximation of the shortest vector problem.

3 LLL Algorithm

The *LLL* algorithm finds a reduced basis for the lattice. The idea is to mimic GS but tranform vectors to those within the lattice. At the same time, we need to keep in mind that vectors don't become too short (to form a reduced basis). The algorithm is very mysterious, we shall first present he algorithm then argue that it halts quickly and also that it works correctly.

Input: A basis $\{b_i\}$.

1: Find the GS basis $\{b_i^\star\}$.
(Reduction Step)
2: **for** $i = 2$ to n **do**
3: **for** $j = i - 1$ to 1 **do**
4: $b_i = b_i - \alpha_{ij} b_j$ where $\alpha_{ij} = \left\lfloor \frac{\langle b_i, b_j^\star \rangle}{\|b_j^\star\|^2} \right\rceil$
5: **end for**
6: **end for**
7: *(Swap Step)* If there exists an i such that

$$\frac{3}{4}\|b_i^\star\|^2 > \left\|b_{i+1}^\star + \mu_{i+1,i} b_i^\star\right\|^2$$

then swap b_i and b_{i+1} and go to step 1.
8: **output** $b_1, b_2, \cdots, b_n$.

3.1 The Reduction Step

The reduction step is basically an approximation to the GS orthogonalization, but staying on the lattice. We shall show that we actually get pretty close to the orthogonal basis.

For the basis $\{b_i\}$, let the GS basis is $\{b_i^\star\}$. If we were to consider the matrix with columns as $\{b_i\}$ as vectors over the GS basis as the standard basis, then B would look like an upper triangular matrix with 1s on the diagonal.

The reduction step makes the other non-diagonal entries small (bounded by 1/2). We shall see how this is achieved.

The two *for* loops are designed cleverly so that you never undo something that you have already done. The key point to note is that in the reduction step, the GS basis is maintained. Look at an intermediate step, say at i, j. By induction, assume that all columns whose index is less than i has already been taken care of.

And since we have gone up to j, the i-th column is fixed from bottom to top. Since the GS basis is fixed, if we had removed the roundoff in α_{ij} when we did $b_i = b_i - \alpha_{ij} b_j$ we would have actually got a vector orthogonal to $b_j^\star$ and hence $b_i j$ would have become 0. But since we are just rounding off, we will atleast reduce that value to 1/2. Note that this works only because the GS basis stays the same throughout.

Now we have fixed the index b_{ij} and we can go on to $b_{i,j-1}$. Thus by induction, we have proved that at the end of the reduction step, we have an uppertriangular matrix with 1s on the diagonal and every non-zero entry is bounded by 1/2.

3.2 The Swap Step

The swap step is like a 'check if reduced basis, else rectify' step. The crucial point is that this step will happen for atmost polynomially many steps. To show this, we will develop a certain value (exponential sized) and show that decreases by a constant factor (3/4) and hence can happen atmost polynomially many times.

For a basis B, define

$$\begin{aligned} D_{B,i} &= \prod_{j=1}^{i} \|b_i^\star\|^2 \\ D_B &= \prod_{i=1}^{n} D_{B,i} \end{aligned}$$

It is easy to see that D_B is a value that is at most exponential. We will show that it goes down by 3/4 each time we swap.

Recall that

$$MB^\star = \begin{bmatrix} 1 & & & \\ \mu_{21} & 1 & & \\ \vdots & & \ddots & \\ \mu_{n1} & \mu_{n2} & \cdots & 1 \end{bmatrix} B^\star = B$$

We could do the same by restricting the above equation to just the first i rows and columns. As a notation, we will write this as

$$B_i = M_i B_i^\star$$

Since M_i is a unimodular matrix,

$$\det(B_i B_i^T) = \det(B_i^\star (B_i^\star)^T) = D_{B_i}$$

Consider the case when you are to do the swap operation between i and $i+1$. Then the basis $B = \{b_1, \cdots, b_{i-1}, b_i, b_{i+1}, \cdots, b_n\}$ will now change to $\hat{B} = \{b_1, \cdots, b_{i-1}, b_{i+1}, b_i, b_{i+2}, \cdots, b_n\}$. The only place where the GS basis will differ will be at the i-th index.

In the original basis B, we would have just $b_i^\star$. But in the other basis $\hat{B}$, it is easy to check that $\hat{b}_i^\star = b_{i+1}^\star + \mu_{i+1,i} b_i^\star$. The other vectors would be the same in both cases.

Thus, clearly,

$$\frac{D_{\hat{B}}}{D_B} = \frac{D_{\hat{B},i}}{D_{B,i}} = \frac{\left\|b_{i+1}^\star + \mu_{i+1,i} b_i^\star\right\|^2}{\|b_i^\star\|^2} \leq \frac{3}{4}$$

And hence the swap step is executed only polynomially many times.

3.3 Correctness

The next thing we need to show is that at the end of the algorithm, we do have a reduced basis. This is an easy observation. Since for all indices

$$\begin{aligned} \frac{3}{4}\left\|b_i^\star\right\|^2 &\leq \left\|b_{i+1}^\star + \mu_{i+1,i} b_i^\star\right\|^2 \\ &= \left\|b_{i+1}^\star\right\|^2 + \mu_{i+1,i}^2 \left\|b_i^\star\right\|^2 \\ &\leq \left\|b_{i+1}^\star\right\|^2 + \frac{1}{4}\left\|b_i^\star\right\|^2 \\ \implies \left\|b_i^\star\right\| &\leq 2\left\|b_{i+1}^\star\right\| \end{aligned}$$

And therefore, we indeed have a reduced basis, and hence solves the approximation of the shortest vector problem.

3.4 Sizes of Numbers

Using the matrices that appeared in the reduction step section, we can show using cramer's rule that the numbers do not become very large.

We leave this as an exercise.

4 Factoring over $\mathbb{Q}$

We will see a sketch of the factoring algorithm, and gaps are left as an assignment. The working is very similar to the bivariate hensel lifting.

The algorithm is as follows:

1. Assume $f(x) \in \mathbb{Z}[x]$ is square free.
2. Pick a small ($O(\log n)$ bits long) prime such that $f(x)$ is square free modulo p.
3. Factor $f = gh \pmod p$. where g is irreducible and monic.
4. Hensel lift the factorization k times to obtain $f = g_k h_k \pmod{p^k}$.
5. Solve the linear equation $\tilde{g} = g_k l_k \pmod{p^k}$ for polynomials $\tilde{g}$ and l_k such that their degree is less than $\deg f$.
6. Output $\gcd(f, \tilde{g})$, if trivial output irreducible.

First catch is the following, does a polynomial necessarily have only small factors? Can there be factors with huge numbers in them? The following bound tells us that we are safe in this area.

Lemma 3 (Mignotte's Bound)**.** *If* $f(x) = a_0 + a_1 x + \cdots + a_n x^n$*, then any root* α *of* f *is such that* $|\alpha| < n \max |a_i|$.

Since all coefficients are symmetric polynomials over the roots, we are in good shape.

For the proof of correctness, we need a suitable bound on k to push the proof of the bivariate case through the same this. But the issue is that, we do not have any bounds on the coefficients of $l_k, \tilde{g}$ to make it work. How do we make sure that the solution to the system of equations is small? Enter LLL.

Look at $\tilde{g} = g_k l_k + p^k r_k$ for any polynomial r. We can easily induce a lattice structure on this by choosing a natural basis. Over this lattice, we can now ask for a short vector. Note that LLL will not give us the shortest vector but a $2^{\frac{n-1}{2}}$ is good enough!

Using that, a bound on k can be fixed and the same proof of bivariate factorization will go through. The gaps are left to the readers to fill in.