LECTURE 09

3/7/2012

Note Title 3/3/2012

TODAY

The LLL (Lenstra, Lenstra, Lovasz) Algorithm

Lattices:

$L \subseteq \mathbb{R}^n$ is a lattice if it is a discrete, additive set.

(1) Discrete: $\forall x \in L \ \exists \delta > 0$ s.t.

$$B(x, \delta) \cap L = \{x\}$$

Ball of radius δ around x

(2) Additive: $\forall x, y \in L$, $x - y \in L$

Aside:

Additivity $\Rightarrow$ can exchange quantifiers in "Discrete". Specifically

$$\exists \delta \text{ s.t. } \forall x \in L \quad B(x, \delta) \cap L = \{x\}.$$

Bases & Representations

Claim: $x_1, \ldots x_{n+1} \in L \subseteq \mathbb{R}^n$ for lattice L

$\Rightarrow \alpha_1 \ldots \alpha_{n+1} \in \mathbb{Z}$ s.t. $\sum \alpha_i x_i = 0$

"**Example**":

$x_1 = 1$, $x_2 = \sqrt{2}$ in $\mathbb{R}^1$

no such $\alpha_1, \alpha_2 \in \mathbb{Z}$. $\Rightarrow$ x_1, x_2 not in any lattice. Not discrete.

Claim: $\forall L \subseteq \mathbb{R}^n \quad \exists x_1 \ldots x_m \in L \quad m \leq n$

x_i's linearly independent s.t.

$$L = \{ \sum \alpha_i x_i \mid \alpha_i \in \mathbb{Z} \}$$

$\{x_1 \ldots x_m\} \stackrel{\Delta}{=}$ Basis of L

Computational Lattices: For most purposes suffices to work with $L \subseteq \mathbb{Z}^n$; ($\approx L \subseteq \mathbb{Q}^n$)

Typically assume L given by basis $\subseteq \{-2^b, \ldots 2^b\}^n$

Shortest Vector Problem

Input: $x_1, \dots x_m \in L = \{ \sum \alpha_i x_i , \alpha_i \in \mathbb{Z} \} \subseteq \mathbb{Z}^n$

Output: $v \in L$ s.t.

$$\|v\|_2 \leq \beta(n) \cdot \|x\|_2 \quad \forall x \in L - \{0\}$$

LLL algorithm achieves $\beta(n) = 2^n$ in polytime.

Suffices for our setting

Recall our problem

Input: $g \in \mathbb{Z}[x]$, degree parameter d,

coeff. bound N, modulus M

Find: $f \in \mathbb{Z}[x]$, $\deg(f) \leq d$

$|\text{coeffs}(f)| \leq N$ s.t.

$\exists\, h \in \mathbb{Z}[x]$ s.t.

$$f = g \cdot h \pmod{M}$$

Claim: Using LLL can either find such f, or claim no solution exists with

$$|\text{coeffs}(f)| \le \frac{N}{d \cdot 2^n}$$

(such a guarantee is good enough for our factorization application)

Proof: Polynomials $\longleftrightarrow$ lattice vectors by coefficients

$$\overset{d+1}{\longleftrightarrow}$$

$$g = \sum_{i=1}^{k} g_i x^i \longrightarrow (g_0, g_1, \ldots g_k, 0, 0, 0) \to V_1$$

$$x \cdot g \longrightarrow (0\ g_0, g_1 \ldots g_k \ldots) \to V_2$$

$$\vdots$$

$$x^{d-k} \cdot g \longrightarrow (0\ 0 - 0\ g_0 \ldots g_k) \to V_{d-k+1}$$

$$+\ \left(\text{mod } m\right) \text{ reductions} \longrightarrow (M\ 0 \ldots 0) \to W_1$$

$$(0\ M\ 0 \ldots 0) \quad \vdots$$

$$\vdots$$

$$(0\ 0 \ldots M) \quad W_{d+1}$$

$\mathbb{Z}$-Span$\{V_1 \ldots V_{d-k+1}, W_0 \ldots W_{d+1}\}$ = poly of form $g \cdot h \pmod m$ $\boxtimes$

SVP Algorithms

For simplicity let $m = n$; can reduce to this case from $m > n$ by some gcd computations ("Hermite normal form") — details omitted.

<u>Warmup</u>: Gauss's algorithm, $n = 2$, $\beta(n) = 1$.

<u>Input</u>: $a, b \in \mathbb{Z}^2$

<u>Alg</u>: 1 $i \leftarrow \arg\min_j \{ \| a - jb \| \}$

$a \leftarrow a - ib$

2 if $\|a\| \le \frac{1}{\sqrt{3}} \|b\|$ swap & goto 1

else output $\min\{\|a\|, \|b\|\}$.

Runtime: Obvious; Every swap shrinks length of $\|b\|$.

Proof of Correctness

Let $V = i \cdot a + j \cdot b$ be shortest vector

Write $a = a^* + \alpha . b$ $\qquad \alpha \in \mathbb{R}$

& $a^* \perp b$ $\qquad |\alpha| \leq \frac{1}{2}$

$$\|V\|^2 = i^2 \|a^*\|^2 + (j-i)^2 \|b\|^2$$

$$\Rightarrow \|V\| \geq i \|a^*\|$$

Since $\|a^*\| > \frac{1}{2} \|a\|$ (1), we must have $i < 2$

$\Rightarrow i = 0$ or 1.

$i = 0 : \quad \Rightarrow \quad j = 1 \Rightarrow V = b$

$i = 1 : \quad \Rightarrow \quad V = a + jb$ but a has minimum length among all such a.

(1) $\|a\| > \frac{1}{\sqrt{3}} \|b\|$ & $\|a\|^2 = \|a^*\|^2 + \alpha^2 \|b\|^2$

$$\Rightarrow \|a^*\|^2 = \|a\|^2 - \alpha^2 \|b\|^2$$

$$\geq \|a\|^2 - 3\alpha^2 \|a\|^2 \geq \frac{1}{4} \|a\|^2$$

$$\Rightarrow \|a^*\| \geq \frac{1}{2} \|a\|$$

(1)

The LLL Algorithm

Idea: Extends Gauss's algorithm to n-dim.

- Main challenges
  - Reduction of b_i w.r.t. $b_1, \ldots b_{i-1}$ is itself intractable
  - Need to do it "heuristically"
  - While still maintaining some approximation guarantees.
  - Choice is subtle; analysis same (not complex)

LLL : Notation

- At any stage has n vectors (basis)

  $b_1 \dots b_n \in L$

- Notation :

  $b_1^*, \dots b_n^* \in \mathbb{R}^n$

  $b_i^* = b_i - \left(\text{projection of } b_i \text{ to space spanned by } b_1 \dots b_{i-1}\right)$

  So b_i^* are orthogonal to each other.

  μ_{ij}'s $\in \mathbb{R}$ are

  such that

  $$b_i = \sum_{j \leq i} \mu_{ij} b_j^* \quad (\text{so } \mu_{ii} = 1)$$

LLL Algorithm

Step 1: "Near Orthogonalization"

- Subtract appropriate multiples of b_j from b_i $(j < i)$ to make sure $-\frac{1}{2} < \mu_{ij} \leq \frac{1}{2}$

  ($\exists$ unique way to do this & requires $\binom{n}{2}$ subtractions)

- Note any change leaves b_i^* invariant.

Step 2: "Swap"

if $\exists$ i s.t. swapping $b_i \longleftrightarrow b_{i+1}$ would reduce b_i^* by factor of $3/4$, do it. Else stop & return b_1.

Running Time

- The amazing potential function Φ

$$\Phi = \prod_{i=1}^{n} Vol_i$$

Where $Vol_i = \prod_{j=1}^{i} b_i^* = \text{Volume}(b_1 .. b_{i-1})$

- Φ is an integer (always), starts at $2^{poly(n,b)}$

- Φ unchanged in step 1; Φ reduces by factor 3/4 in step 2.

$\boxtimes$

Correctness / Performance Guarantee:

- Similar to analysis of Gauss's Algorithm.
- Swap condition $\Leftarrow$?
  - Came from the fact that we can argue b_i^* is not much smaller than b_{i-1}^*
- Note every vector in lattice is at least as long as $\min_i \{ \|b_i^*\| \}$

Lemma: At the end $\|b_i^*\| \geq \frac{1}{2} \|b_{i-1}^*\|$ $\forall i$

Proof: Can write

$$b_i = b_i^* + \mu_{i,i-1} \cdot b_{i-1}^* + a$$

$$b_{i-1} = b_{i-1}^* + b$$

$$a, b \in \text{span}\{b_1, \cdots b_{i-1}\}$$

Swap condition $\Rightarrow$

$$\| b_i^* + \mu \cdot b_{i-1}^* \|^2 \geq \left(\frac{3}{4}\right)^2 \| b_{i-1}^* \|^2$$

$$\Rightarrow \| b_i^* \|^2 \geq \left(\left(\frac{3}{4}\right)^2 - \mu^2\right) \| b_{i-1} \|^2$$

$$= \left(\frac{9-4}{16}\right) \| b_{i-1}^* \|^2$$

$$> \frac{1}{4} \cdot \| b_{i-1}^* \|^2$$

$$\Rightarrow \| b_i^* \| > \frac{1}{2} \cdot \| b_{i-1}^* \|$$

$\boxtimes$

Conclusion:

- Can solve SVP (shortest vector problem) in lattices in ℓ_2 norm, to within 2^n-approx. factor in poly time
- Extends to other norms (all norms within n-factor of each other)
- Can solve CVP (closest vector problem) also to within similar factors.
- Till 1996 SVP was not known to be NP-hard.
- [Ajtai] finally broke through this barrier (NP-hard under randomized reductions)
- Significant hardness of approximation known now; but not expected at $\sqrt{n}$-approximat

- SVP-hardness forms basis of many crypto protocols
- LLL forms basis of many cryptanalytic attacks.
- SVP first hard problem in NP to see some "worst-case" to "average-case" hardness. but not conclusive yet.
- Active area of work....

  ... but first invented for

  ALGEBRA & COMPUTATION !!