Hard Instances of Lattice Problems
Average Case - Worst Case Connections

Christos Litsas

28 June 2012
Outline

Abstract

Lattices

The Random Class

Worst-Case - Average-Case Connection
Hard Problems Already Exist

All Time Classic Hard Problems

- NP-Complete problems
- Factorization
- Discrete Logarithm

reduces to average case: \( \log_{g_2} t = (\log_{g_1} g_2)(\log_{g_1} t)^{-1} \)
Hard Problems Already Exist

<table>
<thead>
<tr>
<th>All Time Classic Hard Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ NP-Complete problems</td>
</tr>
<tr>
<td>▶ Factorization</td>
</tr>
<tr>
<td>▶ Discrete Logarithm</td>
</tr>
</tbody>
</table>

reduces to average case: \( \log_{g_2} t = (\log_{g_1} g_2)(\log_{g_1} t)^{-1} \)

<table>
<thead>
<tr>
<th>Worst-Case Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Those problems are hard only under certain distributions. Often it is not clear how to find such a distribution.</td>
</tr>
</tbody>
</table>
One Step Further

Worst-Case Vs. Average-Case Hardness

A random class of lattices so that if the SVP is easy to solve then the above problems are easy in every lattice.
Lattices
Lattice Definition

**Definition**

Let \( B \in \mathbb{R}^{m \times n} \), we consider the set
\[
\mathcal{L} = \{ y : y = B \cdot x \quad \forall x \in \mathbb{Z}^{1 \times n} \},
\]
that is the set of all integer linear combinations of \( B \). We call every \( \mathcal{L} \) with the above properties a lattice.
Lattices

Figure: An example of a lattice and its basis.
Figure: A lattice has more than one bases.
Properties

- Multiplication by a unimodular matrix produces a new basis.
Properties

- Multiplication by a unimodular matrix produces a new basis.
- Infinite (countable) different bases.
Properties

- Multiplication by a unimodular matrix produces a new basis.
- Infinite (countable) different bases.
- The only part of a lattice that is known is the place where the basis vectors lie.
Definition

Let $\mathcal{L}$ be a lattice, and let a basis for $\mathcal{L}$ is $B = [b_1, \ldots, b_n]$, $b_i$ are the column vectors of $B$, then we define the set

$$\mathcal{P}(B) = \{ y : y = \sum_{i=0}^{n} x_i \cdot b_i, \quad x_i \in [-\frac{1}{2}, \frac{1}{2}) \}.$$  

We call $\mathcal{P}(B)$ the fundamental parallelepiped of $\mathcal{L}$ with respect to the basis $B$. 
Mathematical Tools

▶ Equivalence relation, \( \equiv \mod B \).
Mathematical Tools

- Equivalence relation, \(\equiv \mod B\).
- Efficiently computable distinguished representatives as 
  \(t - B \cdot \lceil B^{-1} \cdot t \rceil\).
Mathematical Tools

- Equivalence relation, $\equiv \mod B$.
- Efficiently computable distinguished representatives as $t - B \cdot \lceil B^{-1} \cdot t \rceil$.
- Partition of the space $\mathbb{R}^n$ by multiplies of fundamental parallelepiped.
Lattice Problems & Solutions

Classic Hard Problems

- P1 approximate SVP
- P2 approximate unique SVP
- P3 approximate SIVP (find a basis)
# Lattice Problems & Solutions

## Classic Hard Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>approximate SVP</td>
</tr>
<tr>
<td>P2</td>
<td>approximate unique SVP</td>
</tr>
<tr>
<td>P3</td>
<td>approximate SIVP (find a basis)</td>
</tr>
</tbody>
</table>

## Classic Algorithms and Bounds

- LLL Reduction Algorithm \(2^{\frac{n-1}{2}} sh(L)\) approximation.
- Babai’s Nearest Plane Algorithm.
# Lattice Problems & Solutions

## Classic Hard Problems

<table>
<thead>
<tr>
<th>P1</th>
<th>approximate SVP</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td>approximate unique SVP</td>
</tr>
<tr>
<td>P3</td>
<td>approximate SIVP (find a basis)</td>
</tr>
</tbody>
</table>

## Classic Algorithms and Bounds

- LLL Reduction Algorithm \(2^{\frac{n-1}{2}} \text{sh}(\mathcal{L})\) approximation.
- Babai’s Nearest Plane Algorithm.

## Bounds

- Shor proved that in LLL the approximation factor can be replaced by \((1 + \epsilon)^n\).
- Minkowski (Convex Body Theorems) \(\text{sh}(\mathcal{L}) \leq c \sqrt{n \det(\mathcal{L})} \frac{1}{n}\).
More on Lattices...

Definition (Dual Lattice)

Let $\mathcal{L}$ be a lattice, we define the dual lattice to be the set $\mathcal{L}^* = \{ y : \forall x \in \mathcal{L} \langle x, y \rangle \in \mathbb{Z} \}$. 

Definition (Smoothing Parameter)
Definition (Dual Lattice)

Let $\mathcal{L}$ be a lattice, we define the dual lattice to be the set
\[ \mathcal{L}^* = \{ y : \forall x \in \mathcal{L} \langle x, y \rangle \in \mathbb{Z} \} . \]

Definition (Smoothing Parameter)

For any $n$-dimensional lattice $\mathcal{L}$ and $\epsilon \in \mathbb{R}^+$, we define its smoothing parameter $\eta_\epsilon(\mathcal{L})$ to be the smallest $s$ such that
\[ \rho_{1/s}(\mathcal{L}^* \setminus \{0\}) \leq \epsilon. \]
Lemma

For any \( s > 0, \ c \in \mathbb{R}^n \) and lattice \( \mathcal{L}(B) \), the statistical distance between \( D_{s,c} \mod \mathcal{P}(B) \) and the uniform distribution over \( \mathcal{P}(B) \) is at most \( \frac{1}{2} \rho_1 / s (\mathcal{L}(B)^* \setminus \{0\}) \). In particular, for any \( \epsilon > 0 \) and any \( s \geq \eta_\epsilon(B) \), holds that

\[
\Delta(D_{s,c} \mod \mathcal{P}(B), U(\mathcal{P}(B))) \leq \epsilon / 2
\]
Definition of $\mathcal{L}$ and $\Lambda$

1. $\mathcal{L}$: $q$-ary lattice.
2. $\Lambda$: the perpendicular lattice of $\mathcal{L}$. 

Definition of $\mathcal{L}$ and $\Lambda$

Symbols

- $B = (u_1 : u_2 : \ldots : u_m), u_i \in \mathbb{Z}^n$.
- Lattice: $\mathcal{L}(B, q) = \{y : y = B \cdot x \mod q, \forall x \in \mathbb{Z}^{1 \times m}\}$ ($\mathcal{L}(B, q) \subseteq \mathbb{Z}^n$).
- Perpendicular Lattice: $\Lambda(B, q) = \{y : y \cdot B \equiv 0 \mod q\}$ ($\Lambda(B, q) \subseteq \mathbb{Z}^n$).

Parameters

- $m = [c_1 n \log n]$
- $q = [n^{c_2}]$
Our Goal

1. Redefine the basis $B$ so that if there is a PT algorithm that finds a shortest vector in $\Lambda$ then it breaks P1, P2, P3 in any lattice.
Substitute $B$ by $\lambda'$

Define $\lambda' = (v_1, \ldots, v_m)$, $v_i \in \mathbb{Z}_q^n$. Every $v_i$ is chosen independently and with uniform distribution from the set of all vectors in $\mathbb{Z}_q^n$. 
First Step

Substitute $B$ by $\lambda'$

Define $\lambda' = (v_1, \ldots, v_m), v_i \in \mathbb{Z}_q^n$. Every $v_i$ is chosen independently and with uniform distribution from the set of all vectors in $\mathbb{Z}_q^n$.

Simultaneous Diophantine Equations

The problem of finding a SV in $\Lambda(\lambda', q)$ is equivalent to solve a linear simultaneous Diophantine equation.
First Step

Substitute $B$ by $\lambda'$

Define $\lambda' = (v_1, \ldots, v_m), \ v_i \in \mathbb{Z}_q^n$. Every $v_i$ is chosen independently and with uniform distribution from the set of all vectors in $\mathbb{Z}_q^n$.

Simultaneous Diophantine Equations

The problem of finding a SV in $\Lambda(\lambda', q)$ is equivalent to solve a linear simultaneous Diophantine equation.

Theorem (Dirichlet)

If $c_1$ is sufficiently large with respect to $c_2$ then there is always a SV in $\Lambda(\lambda', q)$ which is sorter than $n$. 
Problem :-(

$\Lambda(\lambda', q)$ is Unknown to Everybody (Crypto Only)

It seems that there is no way of constructing a shortest vector in $\Lambda(\lambda', q)$. So we don’t have a trapdoor!
Second Step

Substitute $\lambda'$ by $\lambda$

Define $\lambda = (\nu_1, \ldots, \nu_m)$, $\forall i \in \{1, \ldots, m - 1\}$ $\nu_i \in \mathbb{Z}_q^n$ also $\nu_i$ is chosen independently and with uniform distribution from the set of all vectors in $\mathbb{Z}_q^n$. We also define $\nu_m = -\sum_{i=1}^{m-1} \delta_i \nu_i$. Where $\delta_i$ is a, randomly generated, sequence of 0 and 1's.
Second Step

Substitute $\lambda'$ by $\lambda$

Define $\lambda = (v_1, \ldots, v_m)$, $\forall i \in \{1, \ldots, m - 1\} v_i \in \mathbb{Z}_q^n$ also $v_i$ is chosen independently and with uniform distribution from the set of all vectors in $\mathbb{Z}_q^n$. We also define $v_m = - \sum_{i=1}^{m-1} \delta_i v_i$. Where $\delta_i$ is a, randomly generated, sequence of 0 and 1’s.

No Loss of Generality

The distribution of $\lambda$ is exponentially close to the uniform distribution. $\sum_{x \in A} \left| P(\lambda = x) - \frac{1}{A} \right| \leq \frac{1}{2^{cn}}$, where $A$ is the set of all possible values of $\lambda$. 
Worst-Case - Average-Case Connection
Main Theorem

Theorem

There are absolute constants $c_1, c_2, c_3$ so that the following holds:

Suppose that there is a PPT algorithm $A$ which given a value of the random variable $\lambda n, c_1, c_2$ as an input, with a probability of at least $\frac{1}{2}$ outputs a nonzero vector of $\Lambda(\lambda n, c_1, c_2, [n^{c_1}])$ of length at most $n$.

Then, there is a PPT algorithm $B$ with the following properties: If the linearly independent vectors $a_1, \ldots, a_n \in \mathbb{Z}^n$ are given as an input then in polynomial time in $\sum \text{size}(a_i)$ gives the output $(d_1, \ldots, d_n)$ so that with probability of greater than $1 - \frac{1}{2^{\sum \text{size}(a_i)}}$ $(d_1, \ldots, d_n)$ is a basis with $\max \|d_i\| \leq n^{c_3} b l(L)$
Main Tool for the Proof

Easy Construction of a Basis

There is a polynomial time algorithm that from a set of $n$ linearly independent vectors $r_1, \ldots, r_n \in \mathcal{L}$ can construct a basis $s_1, \ldots, s_n$ of $\mathcal{L}$ so that $\max \|s_i\| \leq n \max \|r_i\|$.
Main Tool for the Proof

Easy Construction of a Basis

There is a polynomial time algorithm that from a set of $n$ linearly independent vectors $r_1, \ldots, r_n \in \mathcal{L}$ can construct a basis $s_1, \ldots, s_n$ of $\mathcal{L}$ so that $\max \|s_i\| \leq n \max \|r_i\|$

Defining a new Goal

Construct a set of $n$ linearly independent vectors of $\mathcal{L}$ so that each of them is shorter than $n^{c_3-1} bl(\mathcal{L})$. 
Proof of Main Theorem

Assume that we have the set of linearly independent vectors $a_1, \ldots, a_n \in \mathcal{L}$. Let $M = \max ||a_i||$.
Proof of Main Theorem

Assume that we have the set of linearly independent vectors $a_1, \ldots, a_n \in \mathcal{L}$. Let $M = \max \|a_i\|

First Case (Trivial)

If $M \leq n^{c_3-1} b_l(\mathcal{L})$ we are done.
Proof of Main Theorem

Assume that we have the set of linearly independent vectors $a_1, \ldots, a_n \in L$. Let $M = \max \| a_i \|$

First Case (Trivial)

If $M \leq n^{c_3 - 1} bl(L)$ we are done.

Second Case (Hmmm...)

If $M > n^{c_3 - 1} bl(L)$ we construct (?) a set of linearly independent vectors of $b_1, \ldots, b_n \in L$ so that $\max \| b_i \| \leq \frac{M}{2}$. Then we repeat the algorithm with input the set $b_1, \ldots, b_n$. After $\log_2 M \leq 2 \sum \text{size}(a_i)$ steps we get a set of linearly independent vectors where each of them is shorter than $n^{c_3 - 1} bl(L)$.
\[ \max \| b_i \| \leq \frac{M}{2} \]

1. Starting from the set \( a_1, \ldots, a_n \in \mathcal{L} \) we construct a set of linearly independent vectors \( f_1, \ldots, f_n \in \mathcal{L} \) so that \( \max \| f_i \| \leq n^3 M \) and also the parallelepiped \( W = \mathcal{P}(f_1, \ldots, f_n) \) is very close to a cube.

2. We cut \( W \) into \( q^n \) parallelepipeds each of the form \( \sum t_i f_i + \frac{1}{q} W \), where \( 0 \leq t_i < q \) is a sequence of integers.

3. We take a random sequence of lattice points \( \xi_1, \ldots, \xi_m, m = \lfloor c_1 n \log n \rfloor \) from \( W \). Let \( \xi_j \in \sum \frac{t^{(j)}_i}{q} f_i + \frac{1}{q} W \) then we define \( v_j = (t^{(j)}_1, \ldots, t^{(j)}_n) \).

4. Apply \( A \) to the input \( \lambda' = (v_1, \ldots, v_m) \) and get a vector \( (h_1, \ldots, h_m) \in \mathbb{Z}^n \).

5. Then the vector \( \sum h_j (\xi_j - \eta_j) \in \mathcal{L} \) and its length is at most \( \frac{M}{2} \), where \( \eta_j = \sum \frac{t^{(j)}_i}{q} f_i \).

References


Thank you!!!