Lattice Sieving

Kanav Gupta

IIT Roorkee

August 24, 2020

Shortest Vector Problem

Given a lattice $\mathcal{L}$, defined by a a basis $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2 \ldots \mathbf{b}_n\}$ where $\mathbf{b}_i \in \mathbb{R}^d \ \forall i \in [n]$, find the vector $\mathbf{s} = v_1\mathbf{b}_1 + v_2\mathbf{b}_2 \ldots v_n\mathbf{b}_n$ such that $\|\mathbf{s}\|$ is minimized for $v_i \in \mathbb{Z} \ \forall i \in [n]$.

SVP Solvers

- **Enumeration**
  Recursively travelling all the lattice points within a ball of radius R

SVP Solvers

- **Enumeration**
  Recursively travelling all the lattice points within a ball of radius R
- **Sieving**
  Start with a set S of sampled points in the lattice, and sieve for points such that after every step, size of all points are guaranteed to be a factor $0 < \gamma < 1$ of original sizes.

Sieving

The idea of sieving was first introduced in 2001 by Ajtai et al. (2001), but wasn't formalized and studied much until it was investigated by Nguyen and Vidick (2008) and categorized as practical in asymptotically higher dimensions.
Time complexity of $2^{O(n)}$ of sieving is less than the time complexity of $2^{O(n^2)}$ of enumeration. However, for smaller dimension lattices (less than 50), the time taken by enumeration is much smaller than sieving.

AKS Algorithm

Algorithm 1 The AKS algorithm for the Shortest Vector Problem

Input: An LLL-reduced basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice L satisfying Lemma 3.3, and parameters $0 < \gamma < 1$, $\xi > 0$ such that $\xi = O(\lambda_1(L))$ and $c_0 > 0$.
Output: A shortest vector of L under suitable conditions on (γ, ξ, c_0).

$S \leftarrow \emptyset$
for $j = 1$ to $N = 2^{c_0 n}$ **do**
  $S \leftarrow S \cup \mathsf{sampling}(B, \xi)$ using Algorithm 2.
end for
$R \leftarrow n \max_i \|\mathbf{b}_i\| + \xi$
for $j = 1$ to $k = \lceil \log_\gamma \left(\frac{0.01\xi}{R(1-\gamma)} \right) \rceil$ **do**
  $S \leftarrow \mathsf{sieve}(S, \gamma, R, \xi)$ using Algorithm 3.
  $R \leftarrow \gamma R + \xi$
end for
Compute $\mathbf{v}_0 \in L$ such that $\|\mathbf{v}_0\| = \min\{\|\mathbf{v} - \mathbf{v}'\|$ where $(\mathbf{v}, \mathbf{y}) \in S, (\mathbf{v}', \mathbf{y}') \in S, \mathbf{v} \neq \mathbf{v}'\}$
return $\mathbf{v}_0$.

This algorithm is taken as is from Nguyen and Vidick (2008)

Numbers, what do they mean?

Parameters :

- c_0 - Represents size of the set of sampled points.
- ξ - Perturbation size
- γ - Shrinking Factor

Calculations :

- Line 2 : $N = 2^{c_0 n}$
- Line 5 : $R = n \max \|\mathbf{b}_i\| + \xi$
- Line 6 : $k = \lceil \log_\gamma \left(\frac{0.01\xi}{R(1-\gamma)} \right) \rceil$
- Line 8 : $R = \gamma R + \xi$

Sampling

- We sample not just lattice points. We sample point $\mathbf{y}$ in $B_n(R)$ and then using Babai's rounding algorithm to find a lattice point $\mathbf{v}$ close to $\mathbf{y}$ such that they are at most ξ far.
- As Babai's algorithm garuntees that the the point $\|\mathbf{y}\| \leqslant n \max_i \|\mathbf{b}_i\|$, we have $\|\mathbf{v}\| \leqslant n \max_i \|\mathbf{b}_i\| + \xi$. This is why we set this value to the value of R initially.
- Why do we maintain pairs? - Other than choice making, it is not clear why they have pairs.

Sampling Algorithm

Algorithm 2 Initial sampling

Input: A basis $B = [\mathbf{b}_1, \ldots, \mathbf{b}_n]$ of a lattice L, and a real $\xi > 0$.

Output: A pair $(\mathbf{v}, \mathbf{y}) \in L \times \mathbb{R}^n$ such that $\|\mathbf{y}\| \leq n \max_i \|\mathbf{b}_i\|$ and $\mathbf{y} - \mathbf{v}$ is uniformly distributed in $B_n(\xi)$.

1: $\mathbf{x} \leftarrow_{\text{random}} B_n(\xi)$

2: $\mathbf{v} \leftarrow \mathsf{ApproxCVP}(-\mathbf{x}, B)$ where ApproxCVP is Babai's rounding algorithm [6].

3: $\mathbf{y} \leftarrow \mathbf{v} + \mathbf{x}$

4: **return** $(\mathbf{v}, \mathbf{y})$

Sieving

- Now we have a set of points, we want to iteratively reduce the size of the vectors of the set as well as reduce the size of the set.
- This is achieved by a set of points $C \subset S$ from the original set S called *Centers*. For the rest of the points, we find the point in C that is closest to the point, and replace them with the difference of these two points.

Sieving Algorithm

Algorithm 3 The sieve with perturbations

Input: A set $S = \{(\mathbf{v}_i, \mathbf{y}_i),\ i \in I\} \subseteq L \times B_n(R)$ and a triplet (γ, R, ξ) such that $\forall i \in I,\ \|\mathbf{y}_i - \mathbf{v}_i\| \leq \xi$.

Output: A set $S' = \{(\mathbf{v}'_i, \mathbf{y}'_i),\ i \in I'\} \subseteq L \times B_n(\gamma R + \xi)$ such that $\forall i \in I',\ \|\mathbf{y}'_i - \mathbf{v}'_i\| \leq \xi$.

$C \leftarrow \emptyset$
for $i \in I$ **do**
    if $\exists c \in C\ \|\mathbf{y}_i - \mathbf{y}_c\| \leq \gamma R$ **then**
        $S' \leftarrow S' \cup \{(\mathbf{v}_i - \mathbf{v}_c, \mathbf{y}_i - \mathbf{v}_c)\}$
    else
        $C \leftarrow C \cup \{i\}$
    end if
end for
return S'

Sieving Properties

- Sieving retains perturbations. That is, for each pair $(\mathbf{v}_i', \mathbf{y}_i')$ in output set S', there exists a pair $(\mathbf{v}_i, \mathbf{y}_i)$ in input set S such that $\mathbf{y}_i - \mathbf{v}_i = \mathbf{y}_i' - \mathbf{v}_i'$
- We decide in which center a pair goes only on based on y vector, but we update both of the vectors.

How to choose c_0?

- Suppose we have a value of c_0, then number of sampled points $N = 2^{c_0 n}$, hence the running complexity of the algorithm is $O(N^2)$ and space complexity is $O(N)$. Simple?
- Turns out that the calculation of c_0 is the reason why AKS algorithm was not considered practical.
- Ajtai et al. (2001) followed a very pessimistic approach towards the calculation of this parameter, where there could have been better values.

How to choose c_0?

Let us define some constants which will help us determine the right value of c_0

- c_R - This is the upper limit of the value of c_0, if we just equate the maximum number of points in $\mathcal{L} \cap B_n(R)$ to the number of sampled points. c_R depends on R.
- c_u - This is a representative of the quantity of points which stay inside the ball $B_n(\xi)$ even after adding the shortest vector **s**. Hence, c_u depends on the value of ξ parameter.
- c_s - This is the value calculated because of the shrinking effect. The size limit of the set of centers is calculated to be $2^{c_s n}$. Hence, it depends on γ parameter.

How to choose c_0?

Now we can have 2 goals -

- We want with a high probability that the output set contains a vector shorter than some R_∞.
- We want with a high probability that the output set contains the shortest vector $\mathbf{s}$.

Ajtai et al. (2001) optimize the value of c_0 on the basis of requirements.

Improvements?

Nguyen and Vidick (2008) tried to introduce a heuristic sieving method which addressed the following issues with AKS Algorithm

- They use a lot of packing estimations, which can increase the sample size a lot.
- They do not use birthday paradox to detect collisions in their analyses.
- They remove the non lattice point as a decision maker. So there is no ξ now!

Laarhoven and Mariano (2018) tries to progressively increase the lattice size and it doesn't add new vectors until all the old vectors have "stabilized".

References

Ajtai, M., Kumar, R., and Sivakumar, D. (2001). A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 601–610, New York, NY, USA. Association for Computing Machinery.

Laarhoven, T. and Mariano, A. (2018). Progressive lattice sieving. In Lange, T. and Steinwandt, R., editors, *Post-Quantum Cryptography*, pages 292–311, Cham. Springer International Publishing.

Nguyen, P. and Vidick, T. (2008). Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2:181–207.