6.876 Lecture 8: Micciancio-Voulgaris CVP algorithm

1 The $CVPP$ Problem

The main idea of this lecture is to solve the $CVPP$ (CVP with precomputation) problem. We define $CVPP$ as follows:

Definition 1.1. Given a basis for a lattice L, and unbounded precomputation, we define the $CVPP$ problem as the problem getting as input a point $t \in \mathbb{R}^n$, and returning as an output the $v \in L$ closest to t.

Remark 1.1. *$CVPP$ is known to be NP-hard, while $CVPP_{\sqrt{n}}$ is in P. The gap here is smaller than for CVP.*

Our strategy for solving CVP will involve two parts: first we will solve $CVPP$, and then we will see how to solve CVP given $CVPP$.

2 Voronoi Cell

For the $CVPP$ algorithm, our precomputation will involve computing the *Voronoi cell:*

Definition 2.1 (Voronoi cell)**.** $V(L) = \{x \in R^n : |x| < |x - v| \text{ for all } v \in L\backslash 0\}$ (i.e all points closer to origin than to any other lattice point) similarly, define $\overline{V}(L) = \{x \in R^n : |x| \leq |x - v| \text{ for all } v \in L\backslash 0\}$.

Let H_v be the set of all points that are closer to the origin than to v. We note that $V(L)$ is the intersection of all of the H_v.

We want an efficient way to encode the Voronoi cell of a lattice. Our solution will be to encode the Voronoi cell as the intersection of a finite number of halfspaces H_v.

Definition 2.2 (Voronoi Relevant)**.** $VR \subseteq L$ is *Voronoi relevant* if $V(L) = \cap_{v \in VR} H_v$, and VR is minimal. A vector in VR is called a *Voronoi relevant vector.*

Fact: there exists a Voronoi relevant set VR such that $|VR| \leq 2(2^n - 1)$. In general this is a tight bound.

So how is this related to CVP? Here is our strategy for relating $CVPP$ to CVP (below, by Vor_n we mean the problem of finding a Voronoi relevant subset of L):

1. Reduce Vor_n to CVP_n in 2^n time.
2. Reduce CVP_n to CVP_{n-1} in time $2^{n/2}$.
3. Reduce CVP_{n-1} to Vor_{n-1} in 2^{2n} time (this is where we use the $CVPP$ routine).

In summary: $Vor_n \leq CVP_n \leq CVP_{n-1} \leq Vor_{n-1}$ (where the reductions are in time $2^n, 2^{n/2}, 2^{2n}$, respectively). Note that it's not enough to just use the recursion from the $CVP_n \leq CVP_{n-1}$ reduction, because in the reduction we will have to call CVP_{n-1} many times, so it's more efficient for us to precompute the Voronoi cell once, and then CVP_{n-1} will be solved fast.

We can now solve for the time complexity of CVP_n. We have the recursion $T(n) = 2^n 2^{n/2} 2^{2n} + T(n-1)$, so $T(n) = O(2^{3.5n})$.

3 The $CVPP$ Algorithm

We begin with a proposition, which we will not prove.

Proposition 3.1. *Let $t, t' \in \mathbb{R}^n$, and $t' \in L + t$. Then the following three statements are equivalent:*

1. *t' is a shortest vector in $L + t$.*
2. *$t - t'$ is a closest vector to t in L.*
3. *$t' \in (L + t) \cap \overline{V}(L)$.*

We now present some corollaries to our proposition.

Corollary 3.2. *Suppose $t_1, t_2 \in (L + t) \cap \overline{V}(L)$. Then $|t_1| = |t_2|$.*

Proof. We see that the third condition from our proposition holds for both t_1 and t_2. Therefore, the first condition from our proposition holds, and so both t_1 and t_2 are shortest vectors in $L + t$, and therefore they have the same length. □

Corollary 3.3. *Fix $t \in \mathbb{R}^n$. Then for all $k \in \mathbb{Z}$,*

$$|\{|t'| : t' \in (L + t) \cap \overline{V}(kL)\}| \leq k^n.$$

We will not prove this corollary. We will later invoke this corollary for the case $k = 2$.

The following lemma should solve the problem of finding $t'' \in (L + t) \cap \overline{V}(L)$ given $t' \in (L + t) \cap 2\overline{V}(L)$.

Lemma 3.4. *Pick any $t \notin \overline{V}(L)$. Let $v \in VR(L)$ such that $v = argmax_{v \in VR} \left(\frac{2\langle t,v\rangle}{\langle v,v\rangle}\right)$, and $\alpha = \frac{2\langle t,v\rangle}{\langle v,v\rangle}$. Then the following two statements hold:*

1. *$t - v \in \overline{V}(\alpha L)$.*
2. *$|t - v| < |t|$.*

We will prove the lemma next lecture.

Strategy for the algorithm: we start with t, and then we subtract lattice vectors from t until we land inside the Voronoi cell. With the lemma, we to argue that it will not take too many steps to get into the Voronoi cell. The trick is that at every step we decrease the length of our vector. However, there are at most 2^n possible lengths by Corollary 3.3, so after 2^n steps the algorithm terminates.

We are left to show that we can quickly find a vector to subtract from t that will decrease the size of the vector. For all $t \in \overline{V}(L)$, we have that $|t| \leq |t - v|$ for v as in the lemma. Therefore, $\frac{\langle t,v\rangle}{\langle v,v\rangle} \leq \frac{1}{2}$, and so $\frac{2\langle t,v\rangle}{\langle v,v\rangle} \leq 1$.

For α and v in the lemma, $2\langle t, v\rangle = \alpha\langle v, v\rangle$ and for all $u \neq v, 2\langle t, u\rangle \leq \alpha\langle u, u\rangle$.

We now have an algorithm taking input satisfying $t \in 2\overline{V}(L)$, and outputting $t' \in (L + t) \cap \overline{V}(L)$:

1. Find v out of the Voronoi relevant cells that maximizes $\frac{2\langle t,v\rangle}{\langle v,v\rangle}$.
2. Set $t \leftarrow t - v$.

3. If $t \in \overline{V}(L)$, output t.
4. Else repeat.

The correctness of the algorithm follows from the lemma, and the runtime of the algorithm is $O(|VR| \cdot 2^n) = O(2^{2n})$.

4 $Vor_n \leq CVP_n$

This reduction will rely on the following theorem:

Theorem 4.1 (Voronoi)**.** *$v \in L$ is a Voronoi relevant vector if and only if $\pm v$ are the only shortest vectors in $v + 2L$.*

We do not prove this theorem in this lecture.