6.876 Lecture 9: NP-hardness of (approximate) CVP

Plan for today:

(1) Finish up CVPP
(2) Prove NP-Hardness of (approx) CVP

Finishing up CVPP

Recall from last class: We want a $2^{O(n)}$-time algorithm for CVP. (This is MV '10.) To do this we came up with a CVPP algorithm: Assume we have $2^{O(n)}$ time to do "preprocessing" on the lattice. (In particular, we compute the Voronoi cell $V(L)$.) Then we are given t (and the $V(L)$ we just computed) and asked to find the closest vector to t in L in $2^{O(n)}$ time.

Recall the definition of the *closed Voronoi cell*:

$$\overline{V}(L) = \{x \in \mathbb{R}^n \mid \|x\| \leq \|x - v\| \; \forall v \in L\}$$

We know several facts about $\overline{V}(L)$:

- The number of ($(n-1)$-dimensional) facets is $\leq 2(2^n - 1)$
- The number of "Voronoi relevant" vectors is the number of facets (since half of a VR vector will lie at the center of a facet)

We also have a lemma from last time:

Lemma 1. *For an n-dimensional lattice L and vectors $t, t' \in \mathbb{R}^n$, the following are equivalent:*

- *t' is a shortest vector in the shifted lattice $L + t$*
- *$(t - t')$ is a closest vector to t in L*
- *$t' \in (L + t) \cap \overline{V}(L)$*

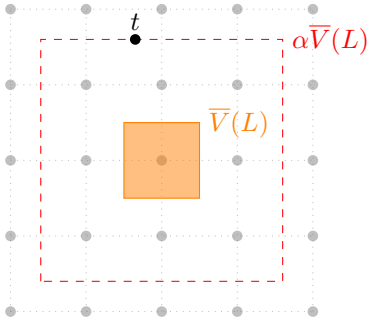
Our goal is to find $t' \in (L + t) \cap \overline{V}(L)$, since by the lemma this will give us a closest vector to t. To do this, we will take the target point t and go on a walk, subtracting lattice vectors from t at each step, until we wanter into the Voronoi cell. Since we start with t and subtract only lattice vectors we always remain in the shifted lattice $L + t$, so once we hit the Voronoi cell we will have a $t' \in (L + t) \cap \overline{V}(L)$.

(As a brief sidenote, this strategy has been around for a long time, but until recently the walk took $O(n^n)$ steps. The new result is a way to walk such that it only takes $2^{O(n)}$ steps.)

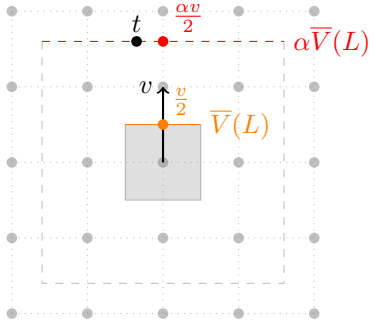
Voronoi Walk. What is the smallest scaling of $\overline{V}(L)$ that contains t? In other words, what is the smallest α such that $t \in \overline{V}(\alpha L)$?

Once we know the α, t will lie on some facet of $\overline{V}(\alpha L)$. Recall that each facet of $\overline{V}(L)$ is centered at $\frac{v}{2}$ for a Voronoi-relevant vector v. If t lies on the facet of $\overline{V}(\alpha L)$ centered at $\frac{\alpha v}{2}$, then we want to move t inward by subtracting v. This is illustrated in the following diagrams:

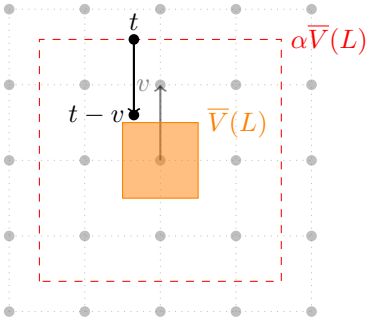
α minimal such that $t \in \overline{V}(\alpha L)$. t is on a facet.



The facet t lies on will be centered at $\frac{\alpha v}{2}$ for some Voronoi-relevant vector v.



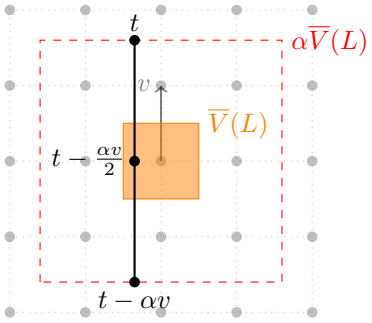
Subtracting v from t will move t inward. By iterating we will eventually land in $\overline{V}(L)$.



We want to show that $t - v$ is still in the scaled Voronoi cell $\alpha\overline{V}(L)$ and that the length $\|t - v\|$ is strictly shorter than $\|t\|$. Since we have from last time a bound on the number of possible lengths of vectors in the Voronoi cell this ensures we are making progress and lets us bound the running time.

Lemma 2. *Let $t \notin \overline{V}(L)$. Let $\alpha = \max_{v \in VR} \frac{2\langle t,v\rangle}{\langle v,v\rangle}$ and let $v = \operatorname{argmax}_{v \in VR} \frac{2\langle t,v\rangle}{\langle v,v\rangle}$. Then $t - v \in \alpha\overline{V}(L)$ and $\|t - v\| < \|t\|$.*

Proof. v is the Voronoi-relevant vector onto which the projection of t is maximal, and α is such that t lies on the facet of $\alpha\overline{V}(L)$ with center $\frac{\alpha v}{2}$. (Note that the facet of $\overline{V}(L)$ with center $\frac{v}{2}$ is given by the equation $\frac{\langle x,v\rangle}{\langle v,v\rangle} = \frac{1}{2}$.)



$\alpha > 1$ (otherwise t would already be in $\overline{V}(L)$), so $t - v$ is somewhere on the segment between t and $t - \alpha v$. $t - \alpha v \in \alpha\overline{V}(L)$, $t \in \alpha\overline{V}(L)$, and $\overline{V}(L)$ is convex. So $t - v \in \overline{V}(L)$.

As for its length, from the diagram it is apparent that as long as $t - v$ lies on the segment between t and $t - \alpha v$ (and not at either endpoint) its length will be strictly shorter than $\|t\|$. Since $\alpha > 1$ this is in fact the case. □

It is worth noting that the above proof works because we are using the Euclidian norm and so $\overline{V}(L)$ is convex. With other norms $\overline{V}(L)$ may not be convex, and the proof may fail.

Recall the following lemma, which was proved last time:

Lemma 3. *There are at most α^n distinct lengths of points in $(t + L) \cap \alpha\overline{V}(L)$.*

This lets us bound the running time.

Complexity, and some open problems. Calculating α seems to require iterating over all Voronoi-relevant vectors $v \in VR$, and there can be up to $2(2^n - 1)$ such vectors. This raises some questions about possible ways to reduce time complexity:

(1) Do we need to enumerate over all $v \in VR$ to find α? This is an open question.
(2) How many steps of walking do we actually need?

There is a very recent result that shows how to walk and reach the Voronoi cell in time polynomial in input length. This is still exponential time overall (otherwise, we would have P = NP, since CVP is NP-hard).

As for space complexity, this algorithm requires 2^n space immediately to store all the Voronoi-relevant vectors. Is it possible to find redundancy in the representation of $\overline{V}(L)$ to reduce space complexity? Improving space complexity is a big open problem that is closely related to integer programming.

Another open question: Can we get space complexity dependent not on $2(2^n - 1)$ but on the actual number of VR vectors, which may be less? (In the algorithm as presented, our space complexity depends not only on the number of VR vectors in our lattice but also on the number of VR vectors in all the sublattices we get by reducing the dimension, and such sublattices may have more VR vectors than the full lattice.)

NP-hardness of (approximate) CVP

Exact CVP. As a warm-up, let's show NP-hardness of exact CVP by a reduction from subset sum. That is, given a subset sum instance $(a_1, \ldots, a_n, S) \in \mathbb{Z}^{n+1}$, we will construct a CVP instance that is equivalent to the subset sum instance. In particular, we want a lattice basis B, a target vector t, and a distance parameter d. (A CVP instance will be a "Yes" instance if there is a vector in $L(B)$ that is d-close to t.)

Our CVP instance is as follows:

$$B = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \\ 2 & & & \\ & 2 & & \\ & & \ddots & \\ & & & 2 \end{bmatrix} \qquad t = \begin{bmatrix} S \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad d = \sqrt{n}$$

We will show that this CVP instance is equivalent to the SSUM instance.

- Suppose our SSUM instance is a YES instance. Let $\vec{x}$ be the solution to the subset sum instance as a vector in $\{0,1\}^n$. Then $B\vec{x} = \begin{bmatrix} S \\ 0 \text{ or } 2 \\ 0 \text{ or } 2 \\ \vdots \\ 0 \text{ or } 2 \end{bmatrix}$. Now $\|B\vec{x} - t\| \leq \sqrt{n}$. So our CVP instance is a YES instance.
- Suppose our CVP instance is a YES instance. Then there is an $\vec{x}$ such that $\|B\vec{x} - t\| \leq \sqrt{n}$. All components of $B\vec{x}$ except the first component are even (because of how we constructed B), so each of the last n component of $B\vec{x}$ will differ from the corresponding component of t by at least 1. This already gives a distance $\|B\vec{x}\| \geq \sqrt{n}$, not even taking the first component into account. So If $\|B\vec{x} - t\| \leq \sqrt{n}$, then $B\vec{x}$ must look like $\begin{bmatrix} S \\ 0 \text{ or } 2 \\ 0 \text{ or } 2 \\ \vdots \\ 0 \text{ or } 2 \end{bmatrix}$, and $\vec{x}$ must be in $\{0,1\}^n$ with $\langle \vec{x}, \vec{a} \rangle = S$. So our SSUM instance is a YES instance.

So CVP is at least as hard as SSUM, and $\text{CVP}_{\sqrt{n}}$ is NP-hard.

Approximate CVP. Let's define $\text{gapCVP}'_\gamma : (B, t, d)$:

YES: $\exists x \in \{0,1\}^n$ such that $\|Bx - t\| \leq d$
NO: $\forall x \in \mathbb{Z}^n$, $\forall w \in \mathbb{Z} \setminus \{0\}$, $\|Bx - wt\| > \gamma d$

This is easier than normal gapCVP: Any YES (resp. NO) instance for gapCVP' is also a YES (resp. NO) instance for gapCVP. We will show that gapCVP'_γ is hard (for some γ).

Exact Set Covering. For $\eta < 1$, we define EXSETCOVER_η as follows: Given subsets $s_1, \ldots s_m \subset [n] = \{1, \ldots, n\}$ and $d \leq m$.

YES: There are at most ηd sets that exactly cover $[n]$ (i.e., each element of $[n]$ appears in exactly one of the subsets.)
NO: Every collection of at most d sets does not cover $[n]$ at all.

Theorem 1 (Goldwasser et al.)**.** *There is an $\eta < 1$ such that exact set covering is NP-hard.*

The theorem will not be proved here.

Let Q be very large. Define an $(n+m)$-by-m matrix B as follows: In the bottom m rows we put the identity. In the top n rows, row j column i is Q if $j \in s_i$ and 0 otherwise. (So the rows represent elements of $[n]$, the columns represent the subsets s_i.)

Let $t = \begin{bmatrix} Q \\ \vdots \\ Q \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, where the top n elements are Q and the bottom m elements are 0. Let $\gamma = \frac{1}{\sqrt{\eta}}$. Let $d' = \sqrt{\eta d}$, where d is the parameter to the EXSETCOVER instance (and d' will be the parameter to the gapCVP' instance.

Claim 1. *With B, d', γ, and t defined as such, this gapCVP$'$ instance is equivalent to the given EXSETCOVER instance.*

Proof.

- Suppose the EXSETCOVER instance is a YES instance. Let x be the vector whose ith component is 1 if s_i is in the cover – since this is a YES instance we can find such an x with Hamming weight $\leq \eta d$. The top n components of Bx are all Q (as are the top n components of t). Of the remaining m components only at most ηd are 1 and the rest are 0. So $\|Bx - t\| \leq \sqrt{\eta d} = d'$.
  So the gapCVP' instance is a YES instance.
- Suppose the EXSETCOVER instance is a NO instance. Then any collection of at most d sets picked will not cover $[n]$. If $x \in \{0,1\}^n$ is the vector corresponding to this collection of sets then one of the top n components of Bx will be zero, and the distance from Bx to any multiple of t will be at least Q, which is large. So no x with Hamming weight less than or equal to d can satisfy $\|Bx - wt\| \leq \gamma d$. On the other

hand, if x has hamming weight greater than d then among the bottom m components of Bx, more than d of them will be 1. So in this case $\|Bx - wt\| \geq \sqrt{d} = \gamma d'$. So this is a NO instance of gapCVP$'$.

□