In this lecture we will see complexity-theoretic lower and upper bounds for SVP and CVP. On the one hand, they are NP-hard (under suitable types of reductions) in their exact versions, and even for small approximation factors. On the other hand, there is good evidence *against* the NP-hardness of their approximate versions for factors $\gamma \geq \sqrt{n/\log n}$.

1 NP-Hardness of CVP and SVP

1.1 NP-Hardness of the Closest Vector Problem

First let us recall the decisional version of the (approximate) Closest Vector Problem.

Definition 1.1 (CVP, decision version). For an approximation factor $\gamma = \gamma(n) \geq 1$, an instance of GapCVP_1 is a basis $\mathbf{B}$ of a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, a target point $\mathbf{t} \in \mathbb{R}^n$, and a distance $d \in \mathbb{R}$. It is a YES instance if $\mathrm{dist}(\mathbf{t}, \mathcal{L}) \leq d$, and is a NO instance if $\mathrm{dist}(\mathbf{t}, \mathcal{L}) > \gamma \cdot d$.

The problem is equivalent to asking whether the coset $\mathbf{t} + \mathcal{L}$ has an element of length at most d or not.

Theorem 1.2 (van Emde Boas [vEB81]). GapCVP_1 *is* NP*-complete.*

Proof. To show that a problem is NP-complete, we need to show that it is in NP, and also that it is NP-hard. The former is easy: a witness w for a YES instance $(\mathbf{B}, \mathbf{t}, d)$ is a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{t} - \mathbf{v}\| \leq d$, which by definition exists for a YES instance and does not exist for a NO instance. Clearly, the conditions can be efficiently verified.[1]

Next we need to show NP-hardness, i.e., we need to give a reduction from some NP-hard problem to GapCVP_1. We reduce from the subset-sum problem, which is a natural choice since it has a very similar linear structure to lattice problems. Recall that the subset-sum problem is: given $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n$ and $S \in \mathbb{Z}$, decide if there exists $\mathbf{x} \in \{0,1\}^n$ such that $\langle \mathbf{a}, \mathbf{x} \rangle = \sum_{i=1}^n a_i x_i = S$. Our reduction takes a subset-sum instance $(a_1, \ldots, a_n, S)$ as input, and outputs the GapCVP_1 instance $(\mathbf{B}, \mathbf{t}, d)$, where

$$\mathbf{B} = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ 2 & & & \\ & 2 & & \\ & & \ddots & \\ & & & 2 \end{pmatrix} \in \mathbb{Z}^{(n+1)\times n}, \quad \mathbf{t} = \begin{pmatrix} S \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{Z}^{n+1}, \quad d = \sqrt{n}.$$

We need to show that the above is a YES instance of GapCVP_1 if and only if the given subset-sum instance is a YES instance. In one direction, suppose the subset-sum instance has a solution $\mathbf{x} \in \{0,1\}^n$. Then for the lattice vector $\mathbf{v} = \mathbf{B}\mathbf{x}$, we have

$$\mathbf{v} - \mathbf{t} = \begin{pmatrix} S \\ 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{pmatrix} - \begin{pmatrix} S \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \pm 1 \\ \pm 1 \\ \vdots \\ \pm 1 \end{pmatrix},$$

[1] One technical subtlety is that the witness must have bit length which is polynomial in the instance length. This holds because the bit length of $\mathbf{v}$ is bounded by the sum of those of $\mathbf{t}$ and d.

so $\|\mathbf{v}-\mathbf{t}\| = \sqrt{n}$ and $(\mathbf{B}, \mathbf{t}, d)$ is a YES instance of GapCVP_1.

In the other direction, suppose that there exists some lattice vector $\mathbf{v} = \mathbf{B}\mathbf{x}$ for $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{v}-\mathbf{t}\| \leq \sqrt{n}$. Since the last n entries of $\mathbf{v}$ are even, the last n entries of $\mathbf{v}-\mathbf{t}$ are odd, and hence must all be ± 1 because $\|\mathbf{v}-\mathbf{t}\| \leq \sqrt{n}$. Therefore, $\mathbf{x} \in \{0,1\}^n$. Moreover, the first entry of $\mathbf{v}-\mathbf{t}$ must be zero (against because $\|\mathbf{v}-\mathbf{t}\| \leq \sqrt{n}$), so $\mathbf{x}$ is a solution to the subset-sum instance. $\square$

The above theorem is presented only for the ℓ_2 norm. It is not difficult to generalize it to the ℓ_p norm for any $p > 1$, including $p = \infty$.

1.2 NP-Hardness of the Shortest Vector Problem

One might wonder whether similar methods can be used to prove that the decisional Shortest Vector Problem (GapSVP_1) is NP-complete. For the ℓ_2 norm, it turns out to be *much more challenging* to show this—in fact, it was not until 1998 that Ajtai showed NP-hardness, but under a *randomized* reduction [Ajt98]. This means that an efficient (possibly randomized) algorithm for GapSVP_1 would imply that $\mathsf{NP} \subseteq \mathsf{RP}$ (but not necessarily that $\mathsf{NP} = \mathsf{P}$). Even today, it is still not known whether GapSVP_1 in the ℓ_2 norm is NP-hard under a *deterministic* reduction!

Here we show a much easier result, that GapSVP_1 is NP-complete in the ℓ_∞ norm (also known as max norm), defined as $\|\mathbf{x}\|_\infty = \max_i |x_i|$.

Definition 1.3. (SVP in ℓ_∞, decision version) For an approximation factor $\gamma = \gamma(n) \geq 1$, an instance of $\mathsf{GapSVP}_\gamma^{(\infty)}$ is a basis $\mathbf{B}$ of a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$ and a distance $d \in \mathbb{R}$. It is a YES instance if the minimum distance of $\mathcal{L}$ in ℓ_∞ norm is at most d, i.e., if $\lambda_1^{(\infty)}(\mathcal{L}) \leq d$, and is a NO instance if $\lambda_1^{(\infty)}(\mathcal{L}) > \gamma \cdot d$.

Theorem 1.4 (van Emde Boas [vEB81]). *$\mathsf{GapSVP}_1^{(\infty)}$ is NP-complete.*

Proof. Membership in NP is easy to see: a witness for instance $(\mathbf{B}, d)$ is a vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ for which $\|\mathbf{v}\|_\infty \leq d$, which can be efficiently checked.

For NP-hardness, we reduce from the NP-hard "weak partition" problem, which is a homogeneous variant of the subset-sum problem. The weak partition problem is: given $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}^n$, determine whether there exist disjoint sets $X, Y \subseteq \{1, \ldots, n\}$, not both empty, such that $\sum_{i \in X} a_i = \sum_{i \in Y} a_i$. Equivalently, it asks whether there is a nonzero $\mathbf{x} \in \{-1, 0, +1\}^n$ such that $\langle \mathbf{a}, \mathbf{x} \rangle = 0$. (The indices of the -1 entries of $\mathbf{x}$ correspond to the elements of X, and the indices of the $+1$ entries correspond to the elements of Y.)

The reduction works as follows: given an instance $\mathbf{a} = (a_1, \ldots, a_n)$ of the weak partition problem, it outputs the following instance $(\mathbf{B}, d)$ of $\mathsf{GapSVP}_1^{(\infty)}$:

$$\mathbf{B} = \begin{pmatrix} 2a_1 & 2a_2 & \cdots & 2a_n \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \in \mathbb{Z}^{(n+1) \times n}, \quad d = 1.$$

We need to show that the given weak partition instance is a YES instance if and only if the above $\mathsf{GapSVP}_1^{(\infty)}$ instance is a YES instance. In one direction, suppose that there exists a nonzero solution $\mathbf{x} \in \{0, \pm 1\}^n$ to the weak partition instance, so that $\langle \mathbf{a}, \mathbf{x} \rangle = 0$. Then the lattice vector $\mathbf{B}\mathbf{x} \in \mathcal{L}(\mathbf{B})$ is nonzero and has ℓ_∞ norm $\|\mathbf{B}\mathbf{x}\|_\infty = 1$, as desired. In the other direction, suppose there exists a nonzero lattice vector $\mathbf{v} = \mathbf{B}\mathbf{x} = \binom{2\langle \mathbf{a}, \mathbf{x} \rangle}{\mathbf{x}} \in \mathbb{Z}^{n+1}$ for $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{v}\|_\infty \leq 1$. The first entry $2\langle \mathbf{a}, \mathbf{x} \rangle$ of $\mathbf{v}$ is even, so it must be zero. The remainder of $\mathbf{v}$ is just the vector $\mathbf{x}$, so we must have $\mathbf{x} \in \{0, \pm 1\}^n$ and $\mathbf{x} \neq \mathbf{0}$. This means that $\mathbf{x}$ is a solution to the weak partition instance, which completes the proof. $\square$

1.3 Other Results

As mentioned above, in 1998 Ajtai [Ajt98] showed that GapSVP_1 in the ℓ_2 norm is NP-complete under a randomized reduction. This has since been substantially improved: over a series of works [Mic98, Kho04, HR07], it has been shown that GapSVP_c in ℓ_2 is NP-complete (still under randomized reduction) for any *constant* approximation factor $\gamma = O(1)$, and even under nearly polynomial factors $\gamma = 2^{\log^{1-\epsilon} n}$ for any constant $\epsilon > 0$ if NP cannot be solved in (randomized) quasi-polynomial $2^{\mathrm{poly}(\log n)}$ time. For CVP, the state of the art is that GapCVP_γ is NP-complete under deterministic reduction for factors as large as $\gamma = n^{\Omega(1/\log\log n)}$ [ABSS93], which is "almost polynomial" in n.

A natural question is, how far might we hope to increase the approximation factors γ for the NP-hardness of GapSVP and GapCVP? There are (at least) two answers:

1. Clearly, we should not expect to have NP-hardness for very large factors $\gamma \geq 2^n$, because GapSVP_γ and GapCVP_γ for such factors can be solved in polynomial time using LLL.

2. More interestingly, we should not expect to have NP-hardness for factors $\gamma \geq \sqrt{n/\log n}$. We will show why this is the case in the next section.

2 The Goldreich–Goldwasser Protocol

Clearly, $\mathsf{GapCVP}_\gamma \in \mathsf{NP}$ for any $\gamma \geq 1$. To show that GapCVP_γ is not likely to be NP-complete for $\gamma \geq \sqrt{n/\log n}$, Goldreich and Goldwasser [GG98] proved that it belongs to the complexity class coAM. That is, the *complement* problem $\mathsf{coGapCVP}_\gamma$—which simply flips the YES and NO instances of GapCVP_γ—is in the class AM of problems that have "Arthur–Merlin protocols," defined below.

The Goldreich–Goldwasser result is significant because if GapCVP_γ was NP-complete for some $\gamma \geq \sqrt{n/\log n}$, then it would follow that $\mathsf{NP} \subseteq \mathsf{coAM}$.[2] It is known that this would imply the collapse of the polynomial-time hierarchy, which is considered very unlikely. Therefore, this can be considered strong evidence (but not proof!) that $\mathsf{GapCVP}_{\sqrt{n/\log n}}$ is not NP-complete.

2.1 $\mathsf{coGapCVP}_{\sqrt{n/\log n}} \in \mathsf{AM}$

Informally, the complexity class AM consists of decision/promise problems for which an unbounded prover can convince an efficient randomized verifier that an instance is a YES instance, but even a (possibly malicious) unbounded prover cannot reliably convince the verifier on a NO instance.

Definition 2.1 (AM)**.** A promise problem $L = (L_{\text{YES}}, L_{\text{NO}})$ is in AM if there exists a constant-round protocol between a probabilistic polynomial-time Turing machine A ("Arthur") and a computationally unbounded Turing machine M ("Merlin") with the following properties:

[2] There are some technical subtleties here related to the fact that GapCVP_γ is a *promise* problem, but the chain of reasoning holds for a wide class of reductions by which GapCVP_γ might be shown NP-complete.

- *Completeness:* for any YES instance $x \in L_{\text{YES}}$, we have that $\Pr[A(x) \leftrightarrow M(x) \text{ accepts}] = 1$, i.e., M always convinces A to accept.
- *Soundness:* for any NO instance $x \in L_{\text{NO}}$ and for any unbounded M^*, we have that $\Pr[A(x) \leftrightarrow M^*(x) \text{ accepts}] \leq 1 - 1/\operatorname{poly}(|x|)$, i.e., A rejects with some noticeable probability.

It is straightforward to show that by repeating the protocol in parallel a polynomial number of times, the "soundness error" (i.e., the probability that A accepts on a NO instance) can be made very small, e.g., 2^{-n}.

Theorem 2.2 (Goldreich–Goldwasser [GG98]). $\mathsf{coGapCVP}_\gamma \in \mathsf{AM}$ *for* $\gamma = \sqrt{n/\log n}$ *(or more generally, any* $\gamma = \Omega(\sqrt{n \log n})$*).*

To prove this theorem, we need to give an Arthur–Merlin protocol which causes Arthur to accept whenever the target point $\mathbf{t}$ is *far* from the given lattice $\mathcal{L}$, i.e., when all the vectors in the coset $\mathbf{t} + \mathcal{L}$ have length more than γd (these are the YES instances of $\mathsf{coGapCVP}$). On the other hand, when the coset $\mathbf{t} + \mathcal{L}$ contains a vector of length at most d (a NO instance of $\mathsf{coGapCVP}$), Arthur should reject with noticeable probability. Note that it's not obvious how to convincingly prove the *absence* of a short vector in a lattice coset; this is where *interaction* with an unbounded prover helps.

The intuition behind the protocol is as follows. Arthur first flips a fair coin. If it comes up heads, he chooses a "uniformly random" point in the lattice $\mathcal{L}$; if it comes up tails, he chooses a "uniformly random" point in the coset $\mathbf{t} + \mathcal{L}$. Let $\mathbf{w}$ denote the resulting point. Arthur then randomly chooses uniform "noise" $\mathbf{e}$ from the ball of radius $(\gamma d)/2$, and sends $\mathbf{x} = \mathbf{w} + \mathbf{e}$ to Merlin. Merlin—who, to recall, is computationally unbounded—is supposed to figure out whether Arthur's coin came up heads or not, i.e., whether $\mathbf{w} \in \mathcal{L}$ or $\mathbf{w} \in \mathbf{t} + \mathcal{L}$. Under what conditions can Merlin always do this, versus necessarily having some noticeable probability of failing?

Notice that if $\operatorname{dist}(\mathbf{t}, \mathcal{L}) \geq \gamma d$, then there is *no overlap* between the balls centered at the points of $\mathcal{L}$ and ones centered at the points of $\mathbf{t} + \mathcal{L}$, so Merlin can always give the correct answer. On the other hand, if $\operatorname{dist}(\mathbf{t}, \mathcal{L}) \leq d$, we will argue that the *overlap* between the two collections of balls is relatively large, hence Merlin must make a mistake with some noticeable probability. We now formalize the protocol and its analysis to prove the theorem. In particular, we eliminate the (mathematically problematic) need for a "uniformly random" lattice point by working *modulo the lattice*, using the fundamental parallelepiped of the input basis as a fundamental region.

Proof of Theorem 2.2. The Arthur–Merlin protocol is as follows. Arthur and Merlin are given some $\mathsf{coGapCVP}$ instance $(\mathbf{B}, \mathbf{t}, d)$ as input. Arthur chooses a bit $b \in \{0, 1\}$ and $\mathbf{e} \leftarrow r\bar{\mathcal{B}}$ uniformly at random, where $r = (\gamma d/2)$ and $\bar{\mathcal{B}}$ is the closed unit ball. Arthur then sends the vector

$$\mathbf{x} := (b \cdot \mathbf{t} + \mathbf{e}) \bmod \mathbf{B}$$

to Merlin, i.e., $\mathbf{x}$ is the unique element of $(b\mathbf{t} + \mathbf{e} + \mathcal{L}(\mathbf{B})) \cap \mathcal{P}(\mathbf{B})$ (which is easy to compute). If $\operatorname{dist}(\mathbf{x}, \mathcal{L}) \leq r$, Merlin returns $b' = 0$; otherwise he returns $b' = 1$. Arthur accepts if $b' = b$.

First we show completeness. Let $(\mathbf{B}, \mathbf{t}, d)$ be a YES instance of $\mathsf{coGapCVP}_\gamma$, so $\operatorname{dist}(\mathbf{t}, \mathcal{L}) > \gamma d$ where $\mathcal{L} = \mathcal{L}(\mathbf{B})$. By the triangle inequality, for *any* $\mathbf{x} \in \mathbb{R}^n$, at most one of $\operatorname{dist}(\mathbf{x}, \mathcal{L}) \leq r$ and $\operatorname{dist}(\mathbf{x} - \mathbf{t}, \mathcal{L}) \leq r$ can hold. When $b = 0$, we have $\mathbf{x} = \mathbf{e} \bmod \mathbf{B}$, so $\operatorname{dist}(\mathbf{x}, \mathcal{L}) = \operatorname{dist}(\mathbf{e}, \mathcal{L}) \leq r$, hence Merlin correctly returns $b' = 0$. Similarly, when $b = 1$, we have $\mathbf{x} = \mathbf{t} + \mathbf{e} \bmod \mathbf{B}$, so $\operatorname{dist}(\mathbf{x} - \mathbf{t}, \mathcal{L}) \leq r$, so $\operatorname{dist}(\mathbf{x}, \mathcal{L}) > r$ and Merlin correctly return $b' = 1$.

Proving soundness is more involved. Let $(\mathbf{B}, \mathbf{t}, d)$ be a NO instance, so $\operatorname{dist}(\mathbf{t}, \mathcal{L}) \leq d$ where $\mathcal{L} = \mathcal{L}(\mathbf{B})$; we need to show that Merlin answers incorrectly with some noticeable $1/\operatorname{poly}(n)$ probability.

To see this, let $\mathbf{v} \in \mathcal{L}$ be a lattice vector for which $\|\mathbf{t}'\| \leq d$ where $\mathbf{t}' = \mathbf{t} - \mathbf{v}$. Now observe that Arthur's message $\mathbf{x}$ in the protocol is *identically distributed* to one generated in a slightly different way in the case $b = 1$, as $\mathbf{x} := \mathbf{t}' + \mathbf{e} \bmod \mathbf{B}$ (the case $b = 0$ is unchanged). This is simply because $\mathbf{t}' + \mathbf{e} \bmod \mathbf{B}$ equals $\mathbf{t} + \mathbf{e} \bmod \mathbf{B}$ for any $\mathbf{v} \in \mathcal{L}$ and $\mathbf{e} \in \mathbb{R}^n$. Now let $I = r\bar{\mathcal{B}} \cap (\mathbf{t}' + r\bar{\mathcal{B}})$ be the intersection of the balls centered at the origin and at $\mathbf{t}'$, and observe that for any $\mathbf{y} \in I$ (even before reducing modulo $\mathbf{B}$), it is equally likely that Arthur chose $b = 0$ and $\mathbf{e} = \mathbf{y}$ versus $b = 1$ and $\mathbf{e} = \mathbf{y} - \mathbf{t}'$.[3] So, in this case Merlin cannot do any better than a random guess, which succeeds with probability $1/2$. Therefore, the probability that Merlin answers incorrectly is at least half of

$$\frac{\operatorname{vol}(r\bar{\mathcal{B}} \cap (\mathbf{t}' + r\bar{\mathcal{B}}))}{\operatorname{vol}(r\bar{\mathcal{B}})}. \tag{2.1}$$

Therefore, it suffices to give a lower bound on the above quantity, which by rescaling is the fraction of overlap between two n-dimensional balls of unit radius whose centers are $\delta \leq 2/\gamma$ apart. It is not hard to see that the intersection contains a cylinder C with radius $\sqrt{1-\delta^2}$ and height δ. The volume of an n-dimensional unit ball is known to be

$$V_n = \frac{\pi^{n/2}}{(n/2)!},$$

where the generalized factorial function satisfies $0! = 1$, $n! = n(n-1)!$ for all real $n \geq 1$, and $(1/2)! = \sqrt{\pi}$. We also need the fact that $(n + \frac{1}{2})!/n! = \Theta(\sqrt{n})$.

Therefore, the quantity in Equation (2.1) is at least

$$\begin{aligned}\frac{\delta \cdot (1-\delta^2)^{(n-1)/2} \cdot V_{n-1}}{V_n} &= \frac{\delta \cdot (1-\delta^2)^{(n-1)/2} \cdot \pi^{(n-1)/2} \cdot (n/2)!}{(n/2 - 1/2)! \cdot \pi^{n/2}} \\ &= (1-\delta^2)^{(n-1)/2} \cdot \Theta(\delta\sqrt{n}).\end{aligned}$$

Recalling that $(1 - 1/n)^n = 1/O(1)$ (indeed, it approaches $1/e$ as n grows), we have $(1 - (\log n)/n)^n = 1/O(1)^{\log n} = 1/\operatorname{poly}(n)$. So for any $\delta = O(\sqrt{(\log n)/n})$ and hence any $\gamma = \Omega(\sqrt{n/\log n})$, the above quantity is $1/\operatorname{poly}(n)$, as needed. □

2.2 Summary

To summarize:

- $\mathsf{GapCVP}_{n^{1/\log\log n}}$ is NP-complete.
- $\mathsf{GapCVP}_{\sqrt{n/\log n}} \in \mathsf{coAM}$, so it is unlikely to be NP-hard.
- A work of Aharonov and Regev [AR04] (which we will cover later in this course) showed that $\mathsf{GapCVP}_{\sqrt{n}}$ is in coNP, and hence is unlikely to be NP-hard.
- GapCVP_{2^n} is in P, due to the LLL algorithm.

[3] This is not quite rigorous, because we are conditioning on an event of probability zero. This can be fixed by instead using measure.

References

[ABSS93] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. Preliminary version in FOCS 1993. Page 3.

[Ajt98] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19. 1998. Pages 2 and 3.

[AR04] D. Aharonov and O. Regev. Lattice problems in NP $\cap$ coNP. *J. ACM*, 52(5):749–765, 2005. Preliminary version in FOCS 2004. Page 5.

[GG98] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000. Preliminary version in STOC 1998. Pages 3 and 4.

[HR07] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *STOC*, pages 469–477. 2007. Page 3.

[Kho04] S. Khot. Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005. Preliminary version in FOCS 2004. Page 3.

[Mic98] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2000. Preliminary version in FOCS 1998. Page 3.

[vEB81] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, University of Amsterdam, 1981. Pages 1 and 2.