

HOMEWORK 4

Instructor: Daniele Micciancio

UCSD CSE

In this assignment we explore lattices which are very close to achieving Minkowski's bound, first using an iterative improvement process, and then considering the random lattices used in lattice-based cryptography.

1 Almost Perfect Lattices

- a) Prove that for any n , there is an n -dimensional lattice $\Lambda \subset \mathbb{R}^n$ such that any point $\vec{t} \in \mathbb{R}^n$ is within (Euclidean) distance $1.5 \lambda_1(\Lambda)$ from Λ , i.e., the covering radius of Λ is at most $\mu(\Lambda) \leq 1.5 \lambda_1(\Lambda)$. [Hint: follow the same approach described in the lecture notes to prove the weaker bound $\mu(\Lambda) \leq 2 \lambda_1(\Lambda)$, and adapt it to the problem here using vectors for the form $\frac{1}{3} \vec{v}$ instead of $\frac{1}{2} \vec{v}$.]
- b) Show that the lattice constructed in part (a) is within a constant factor from Minkowski's bound, i.e., its minimum distance satisfies $\lambda_1 \in [\frac{1}{c} B, B]$ for some constant $c > 1$ where $B = \sqrt{n} \det^{1/n}(\Lambda)$. What is the smallest constant you can deduce from the construction in part (a)?

2 Minkowski's bound for random lattices

Let $A = [I, \bar{A}] \in \mathbb{Z}_q^{k \times n}$, where I is the identity matrix, and $\bar{A} \in \mathbb{Z}_q^{k \times (n-k)}$. To simplify computations, in this problem we use the minimum distance λ_1^∞ in the ℓ_∞ norm $\|\vec{x}\|_\infty = \max_i |x_i|$.

- a) Give a basis for the lattice $\Lambda_q^\perp(A) = \{\vec{x} \in \mathbb{Z}^n: A\vec{x} \equiv \vec{0} \pmod{q}\}$ and prove that your solution is correct.
- b) Use Minkowski's theorem to give an upper bound $\lambda_1^\infty(\Lambda_q^\perp(A)) \leq B$ in terms of k, n, q .
- c) Prove that there is a constant $c > 0$ (independent of n, k, q) such that $\lambda_1^\infty(\Lambda_q^\perp(A)) \geq \frac{1}{c} B$ with high probability when A' is chosen uniformly at random. ("High probability" means very close to 1 when n, m, q are sufficiently large.) [Hint: use a union bound over all candidate short vectors.]

3 Default Project

This is not part of the homework assignment, it is just an idea for a possible project. If you have already started working on a project, please send an email to the instructor with a brief description of what you are working on. (E.g., a project title, list of team members if working on a group project, any pointers to relevant papers you are reading, and what you are planning to do, etc.) Otherwise, if you have not yet come up with a project idea, you can use the following as a default project.

Lattices close to Minkowski's bound make good error correcting codes, but in order to be algorithmically useful you need an efficient way to solve the closest vector problem on them. (Here the lattice is fixed, but the CVP target can still be arbitrarily chosen and is considered as the input to CVP.)

The random lattices described in Problem 2 are close to Minkowski's bound, but they are hard cryptographic lattices: solving CVP on such random lattices is as hard as approximating various lattice problems in the worst-case!

So, consider the solution to Problem 1. (For the purpose of this project, you may use the simpler construction using $\vec{v}/2$ described in the lecture notes.) While not necessarily efficient, the construction of these lattices is algorithmic: it involves picking a target point $\vec{h}$ far from the lattice (ideally, a deep hole), finding a lattice point $\vec{v}$ close to $2\vec{h}$, and adding $\vec{v}/2$ to the lattice. If you start from a lattice Λ where CVP can be solved efficiently (e.g., $\mathbb{Z}^n$), this can be done efficiently. (For the selection of $\vec{h}$, you can pick a random point, and use the CVP algorithm for Λ to check that it is far from the lattice. By triangle inequality a random $\vec{h}$ will be at distance at least $\mu/2$ from the lattice.) Moreover, CVP in the new lattice $\Lambda \cup \left(\Lambda + \frac{\vec{v}}{2}\right)$ can still be solved efficiently (with double the running time) by making two calls to the CVP algorithm for Λ .

Naturally, as the time to solve CVP this way doubles at every iteration, the time required to solve CVP (and build a dense lattice) in this naive way quickly becomes infeasible. Can you find a better way to build dense lattices and solve CVP in them? You can play with the choice of the initial lattice Λ , and the selection of $\vec{v}$ at every step. (Perhaps you can find an explicit/deterministic choice of vectors $\vec{v}$ for which CVP can be solved efficiently, at least for the type of targets $2\vec{h}$ used by the construction.)

These are just ideas to get started. The project is open ended, so feel free to explore variants, or alternative ideas. You can approach the problem theoretically (trying to prove bounds on the performance of your lattice/algorithm) or experimentally (implementing your method and measuring your performance.) Alternatively, if you are more into reading papers, you can explore the literature, and search for papers providing explicit (or at least algorithmically efficient) constructions of dense lattices, and explore the algorithmic solution of CVP in these lattices. It is up to you to decide. This is a project, choosing what to do and how much is left to your initiative. Keep in mind that the goal is not to solve any specific problem, but to do something interesting and learn something in the process.