Lattices

- **Definition:** Let \( b_1, \ldots, b_n \in \mathbb{R}^m \) be linearly independent. The lattice they generate is 
\[
L(b_1, \ldots, b_n) = \{x_i b_i | x_i \in \mathbb{Z}\}
\]
- If \( B \) is an \( m \times n \) matrix with \( b_i \) as cols, 
\[
L(B) = L(b_1, \ldots, b_n) = \{Bx | x \in \mathbb{Z}^n\}
\]
- **Rank** of lattice is \( n \)
- **Dimension** of lattice is \( m \)
- **Full-rank** if \( n=m \)
- Alternatively:
  - **Definition:** \( L \subseteq \mathbb{R}^m \) is a lattice if:
    - \( x, y \in L \Rightarrow x+y \in L \) and \(-x \in L\)
    - \( L \) is a discrete set
  - So could just say a lattice is a discrete subgroup of \((\mathbb{R}^n,+)\)
  - Basis is a minimal set of generators

Today

- **Start lattices**
  - Nice notes (from which we draw extensively) on websites for classes by Madhu Sudan and Oded Regev
- **Next lecture(s): lattice basis reduction**
  - Considered by many to be one of the greatest algorithms of all time
Over reals, can multiply a basis by any invertible matrix and get a new one.
Doing this to a basis for a lattice will usually give a different lattice.
When does multiplying by a matrix $M$ give a new basis for the same lattice?
- Call two bases for same lattice “equivalent”

Can get problems in two ways:
- Could get points that aren’t in $L$
- Could fail to get every point in $L$

When do each of these happen?

- **Definition:** An $n \times n$ matrix $M$ is *unimodular* if all entries are integers and $\det(M) = \pm 1$
- Will prove a few properties (on blackboard)
- **Claim:** $U$ is unimodular iff $U^{-1}$ is unimodular
- **Corollary:** Inverse of unimodular matrix has integer entries
- **Corollary:** Two bases $B_1$ and $B_2$ are equivalent iff $B_2 = UB_1$ for some unimodular $U$
- **Corollary:** Two bases are equiv. iff can get one from other by following ops:
  - $b_i \leftrightarrow b_j + k b_p$, for some $k \in \mathbb{Z}$
  - $b_i \leftrightarrow b_j$ (swap)
  - $b_i \leftrightarrow -b_i$
- **Definition-Theorem:** Let $L = L(B)$ be full-rank. The determinant of $L = \vol(P(B)) = |\det(B)|$, and this is indep of choice of basis.
- Det is inversely proportional to density of lattice
- For non-full-rank, can use lower-dim volumes, and $(\det(B^T B))^{1/2}$
Dual Lattices

- Can define a few equivalent ways:
- Let $\Lambda = L(B)$ be a full-rank lattice
  - Need to be careful if not full-rank
- **Definition 1:** Let $\Lambda^* = \{x \in \mathbb{R}^n | x \cdot v \in \mathbb{Z} \text{ for all } v \in L\}$
- Looks a lot like def. of polar
  - Why didn’t we say $x \cdot v = o \leq \geq 1$?
- **Definition 2:** Let $\Lambda^*$ be the set of all linear functionals from $\Lambda \rightarrow \mathbb{Z}$
  - How would we represent this as a subset of $\mathbb{R}^n$?
- **Definition 3:** For any basis $B$, let $B^* = (B^T)^{-1}$. Let $\Lambda^* = L(B^*)$
  - Called the dual basis to $B$
  - Not obviously basis independent or the same as above
- **Definition 4:** Let $D$ be any matrix s.t. $BD \in \mathbb{Z}^n$ and $\det(B^T D) = \pm 1$. Let $\Lambda^* = L(D)$.

Shortest Vectors and Successive Minima

- Let $\lambda_1$ = length shortest vector in $L$ (in $l_2$ norm)
- What’s the right notion of “second-shortest”?
- The $i^{th}$ successive minimum of $L$ is given by
  $$\lambda_i = \inf \left\{ r | \dim(\text{span}(L \cap B(0, r))) \geq i \right\}$$
- Looks like Ritz-Rayleigh for eigenvalues

Figure 5: $\lambda_1(\Lambda) = 1, \lambda_2(\Lambda) = 2.3$

Dual Lattices

- What to do when not full rank?
- What is determinant of dual lattice?
- What is dual of the dual?
Blichfeld’s Theorem

- **Theorem:** Let:
  - $L$ be full-rank
  - $S \subseteq \mathbb{R}^n$ be any measurable set with $\text{vol}(S) > \det(L)$.
Then there exist (unequal) $z_1, z_2 \in S$ s.t. $z_1 - z_2 \in L$.
- See blackboard and this picture.

![Figure 6: Blichfeld’s theorem](image)

Minkowski’s Theorem(s)

- **Theorem:** Let $L$ be full rank, and let $K$ be any centrally-symmetric convex body s.t. $\text{vol}(K) > 2^n \det(L)$.
  Then $K$ contains a nonzero point of $L$.
- Why do we need the $2^n$?
- See blackboard and this picture.

![Figure 7: Minkowski’s convex body theorem](image)

- **Corollary:** For any full-rank $L$,
  $\lambda_1(L) \leq \sqrt{n} \left( \det(L) \right)^{1/n}$
- How tight is this? And when?

Another Minkowski Theorem

- Can we use the same ideas to bound other successive minima?
- **Theorem:** For any full-rank $L$,
  $\left( \prod_{i=1}^{n} \lambda_i(L) \right)^{1/n} \leq \sqrt{n} \left( \det(L) \right)^{1/n}$
- See blackboard and this picture.

![Figure showing algorithmic questions](image)

Algorithmic Questions

- A bunch of algorithmic questions we can now ask.
- All lattices will have integer coords for these.
  - Same as giving them rational coords. (Why?)
- **Shortest Vector Problem (SVP):** Find the shortest vector in $L$.
  - Could also just ask for length, but this is equiv.
- **Closest Vector Problem (CVP):** Find the vector in $L$ closest to some given point $p$.
  - Both are NP-hard, so ask approx. versions.
- In approx. versions, could ask three types of questions:
  - **Search:** Find a vector within $\gamma$ of optimum.
  - **Optimization:** Approximate the length within a factor of $\gamma$.
  - **Promise:** I promise you that shortest vector is either less than some $r$ or greater than $\gamma r$. Distinguish the two cases.
- No longer all equivalent!
How Hard Are These Problems?

- Very large gaps between best known upper and lower bounds.
- Best poly-time algorithms for approximate SVP and CVP get approx. factors essentially exponential in \( n \)
  - I think the best is something like \( 2^{O(n \log \log n / \log n)} \)
- Best exact algorithm runs in to \( 2^{O(n)} \)
- **Can’t find vector guaranteed by Minkowski!**
- SVP hard to approx. within any constant factor unless \( \text{NP} = \text{RP} \)
- CVP hard to approx. within \( n^{O(1/\log \log n)} \)
- Approx. within \( n^{1/2} \) factor in \( \text{NP} \cap \text{co-NP} \)

Worst-case/average-case equivalence:

- Lattice problems have an amazing property that really nothing else is known to have
- Can show that if they are hard to sufficiently well approximated in worst-case, then there exist average-case hard one-way functions
- Very useful for cryptography
- Pretty surprising