# Introduction to lattices

Let $b_1, \ldots, b_r \in \mathbb{R}^n$ be linearly independent over $\mathbb{R}$.
Consider the following $\mathbb{Z}$-module $\subset \mathbb{R}^n$

$$L := \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_r.$$

Such $L$ is called a *lattice* with basis $b_1, \ldots, b_r$.

**Lattice reduction (LLL):** Given a "bad" basis of $L$, compute a "good" basis of $L$.

What does this mean? Attempt #1: $b_1, \ldots, b_r$ is a "bad basis" when $L$ has another basis consisting of much shorter vectors.

However: To understand lattice reduction, it does not help to focus on lengths of vectors. What matters are: *Gram-Schmidt lengths*.

# Gram-Schmidt

$L = \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_r$

Given $b_1, \ldots, b_r$, the Gram-Schmidt process produces vectors $b_1^*, \ldots, b_r^*$ in $\mathbb{R}^n$ (not in $L$!) with:

$$b_i^* := b_i \quad \text{reduced mod} \quad \mathbb{R}b_1 + \cdots + \mathbb{R}b_{i-1}$$

i.e.

$$b_1^*, \ldots, b_r^* \text{ are orthogonal}$$

and

$$b_1^* = b_1$$

and

$$b_i^* \equiv b_i \quad \text{mod prior vectors.}$$

# Gram-Schmidt, continued

$b_1, \ldots, b_r$: A basis (as $\mathbb{Z}$-module) of $L$.

$b_1^*, \ldots, b_r^*$: Gram-Schmidt vectors (not a basis of $L$).

$b_i^* \equiv b_i$ mod prior vectors

$||b_1^*||, \ldots, ||b_r^*||$ are the *Gram-Schmidt lengths* and
$||b_1||, \ldots, ||b_r||$ are the *actual lengths* of $b_1, \ldots, b_r$.

G.S. lengths are far more informative than actual lengths, e.g.

$$\min\{||v||, \quad v \in L, v \neq 0\} \geqslant \min\{||b_i^*||, \quad i = 1 \ldots r\}.$$

G.S. lengths tell us immediately if a basis is bad
(actual lengths do not).

# Good/bad basis of $L$

We say that $b_1, \ldots b_r$ is a *bad basis* if $||b_i^*|| \ll ||b_j^*||$ for some $i > j$.

Bad basis = later vector(s) have much smaller G.S. length than earlier vector(s).

If $b_1, \ldots, b_r$ is bad in the G.S. sense, then it is also bad in terms of actual lengths. We will ignore actual lengths because:

- The actual lengths provides no obvious strategy for finding a better basis, making LLL a mysterious black box.
- In contrast, in terms of G.S. lengths the strategy is clear:

  (a) Increase $||b_i^*||$ for large $i$, and
  (b) Decrease $||b_i^*||$ for small $i$.

Tasks (a) and (b) are equivalent because $\det(L) = \prod_{i=1}^{r} ||b_i^*||$ stays the same.

# Quantifying good/bad basis

The goal of lattice reduction is to:
(a) Increase $||b_i^*||$ for large $i$, and
(b) Decrease $||b_i^*||$ for small $i$.

Phrased this way, there is a an obvious way to measure progress:

$$P := \sum_{i=1}^{r} i \cdot \log_2(||b_i^*||)$$

Tasks (a),(b), improving a basis, can be reformulated as:

- Moving G.S.-length forward, in other words:
- Increasing $P$.

# Operations on a basis of $L = \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_r$

**Notation:** Let $\mu_{ij} = (b_i \cdot b_j^*)/(b_j^* \cdot b_j^*)$ so that

$$b_i = b_i^* + \sum_{j<i} \mu_{ij}\, b_j^* \qquad (\text{recall} : b_i \equiv b_i^* \bmod \text{prior vectors})$$

LLL performs two types of operations on a basis of $L$:

(I) Subtract an integer multiple of $b_j$ from $b_i$ (for some $j < i$).

(II) Swap two adjacent vectors $b_{i-1}, b_i$.

Deciding which operations to take is based solely on:

- The G.S. lengths $||b_i^*|| \in \mathbb{R}$.
- The $\mu_{ij} \in \mathbb{R}$ that relate G.S. to actual vectors.

These numbers are typically computed to some error tolerance $\epsilon$.

# Operations on a basis of $L = \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_r$, continued

**Operation (I):** Subtract $k \cdot b_j$ from $b_i$ $(j < i$ and $k \in \mathbb{Z})$.

1. No effect on: $b_1^*, \ldots, b_r^*$
2. Changes $\mu_{ij}$ by $k$ (also changes $\mu_{i,j'}$ for $j' < j$).
3. After repeated use: $|\mu_{ij}| \leqslant 0.5 + \epsilon$ for all $j < i$.

**Operation (II):** Swap $b_{i-1}, b_i$, but only when (Lovász condition)

$$p_i \;:=\; \log_2 ||\text{new } b_i^*|| - \log_2 ||\text{old } b_i^*|| \;\geqslant 0.1$$

1. $b_1^*, \ldots, b_{i-2}^*$ and $b_{i+1}^*, \ldots, b_r^*$ stay the same.
2. $\log_2(||b_{i-1}^*||)$ decreases and $\log_2(||b_i^*||)$ increases by $p_i$
3. **Progress counter** $P$ increases by $p_i \geqslant 0.1$.

# Lattice reduction, the LLL algorithm:

**Input:** a basis $b_1, \ldots, b_r$ of a lattice $L$

**Output:** a good basis $b_1, \ldots, b_r$

Step 1. Apply operation (I) until all $|\mu_{ij}| \leqslant 0.5 + \epsilon$.
Step 2. If $\exists_i\ p_i \geqslant 0.1$ then swap $b_{i-1}, b_i$ and return to Step 1.
Otherwise the algorithm ends.

Step 1 has no effect on G.S.-lengths and $P$. It improves the $\mu_{ij}$ and $p_i$'s. A swap increases progress counter

$$P = \sum i \cdot \log_2(||b_i^*||)$$

by $p_i \geqslant 0.1$, so

$$\begin{aligned} \#\text{calls to Step 1} &= 1 + \#\text{swaps} \\ &\leqslant 1 + 10 \cdot (P_{\text{output}} - P_{\text{input}}). \end{aligned}$$

# Lattice reduction, properties of the output:

LLL stops when every $p_i < 0.1$. A short computation, using $|\mu_{i,i-1}| \leqslant 0.5 + \epsilon$, shows that

$$||b^*_{i-1}|| \leqslant 1.28 \cdot ||b^*_i||$$

for all $i$. So later G.S.-lengths are not much smaller than earlier ones; the output is a *good basis*.

# Using LLL to solve (or partially solve!) a problem

LLL solves many problems. Suppose a vector $v$ encodes the solution of a problem, and we construct $b_1, \ldots, b_r$ with

$$v \in \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_r$$

**Solving a problem with a single call to LLL:** If every vector outside of $\mathbb{Z}v$ is much longer than $v$, then the first vector in the LLL output is $\pm v$. The original LLL paper factors $f \in \mathbb{Z}[x]$ by constructing the coefficient vector $v$ of a factor in this way.

**Partial reduction in the combinatorial problem:** If $||b_i^*|| > ||v||$ for all $i \in \{k+1, \ldots, r\}$ then

$$v \in \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_k.$$

The initial basis is usually bad, i.e. $||b_r^*||$ is small: We need LLL to make $||b_r^*|| >$ an upper bound for $||v||$.