Lattice Basis Reduction

LLL algorithm (Lenstra, Lenstra, Lovász)

**INPUT:** \( b_1, \ldots, b_n \in \mathbb{R}^n \text{ lin. independent} \)

**OUTPUT:** \( \tilde{b}_1, \ldots, \tilde{b}_n \in \mathbb{R}^n \) s.t. \( \tilde{B} \) is also a basis

\[ A = \mathbb{Z}(b_1, \ldots, b_n) \]

\[ M = \text{maximum absolute value of an entry of } \tilde{b}_1, \ldots, \tilde{b}_n \]

\[ \gamma(\tilde{b}_1, \ldots, \tilde{b}_n) \leq 2^{n(n-1)/4} \]

\[
\begin{align*}
1. & \text{ Normalize } b_1, \ldots, b_n \\
2. & \text{ Construct GSO vectors } b_1^*, \ldots, b_n^* \not\parallel b_1, \ldots, b_n. \\
3. & \text{ Check if } \| b_i^* \|_2 < \frac{1}{2} \| b_{i-1}^* \|_2 \quad i = 2, \ldots, n \\
& \text{ If no such } i \text{ exists, then } \text{STOP.} \\
& \text{ Else swap } b_i \text{ and } b_{i-1} \text{ in the basis.} \\
& \text{ Go back to step 1.}
\end{align*}
\]
If we assume the algorithm stops with \( k = 2 \), then \( \|b_k\| = 2 \).
\[
\begin{align*}
\lambda & \leq \frac{1}{4} \left( \sum_{j=1}^{i-1} 2^{-j} \| b_i^* \| \right) + \| b_i^* \|^2 \\
& = \sum_{j=1}^{i-1} 2^{-j-2} \| b_i \|^2 + \| b_i^* \|^2 \\
& = \| b_i \|^2 \left( 1 + \sum_{j=1}^{i-1} 2^{-j-2} \right) \\
& = \| b_i \|^2 \left( 1 + \frac{1}{2} + 1 + 2 + \ldots + 2^{i-3} \right) \\
& = \| b_i \|^2 \left( \frac{2}{1-2^{-1}} \right) \\
& \leq 2^{i-1} \| b_i \|^2 \\
\Rightarrow \quad \frac{\| \tilde{b}_i \|^2}{\| b_i \|^2} & \leq 2^{i-1} \\
\Rightarrow \quad \prod_{i=1}^{n} \frac{\| \tilde{b}_i \|^2}{\| b_i \|^2} & \leq \prod_{i=1}^{n} (2^{i-1}) = 2^{\frac{n(n-1)}{2}} \\
\Rightarrow \quad \prod_{i=1}^{n} \frac{\| \tilde{b}_i \|}{\| b_i \|^2} & \leq 2^{\frac{n(n-1)}{4}}
\end{align*}
\]
\[ p(B) = \frac{1}{4} \sum_{i} b_i \]
\[
\begin{align*}
&\leq \frac{3}{4} \| b_{i-1}^* \|^2 \\
\text{Let } \alpha \in \mathbb{R} \text{ s.t. } \\
\| b_{i-1}^* \|^2 = \alpha \| b_{i-1}^* \|^2, \quad \alpha \leq \frac{3}{4} \\
\| b_1^* \| \ldots \| b_n^* \| = \det(B) = \det(B^t) = \| b_1^* \| \ldots \| b_n^* \| \\
\Rightarrow \| b_i^* \|^2 = \frac{1}{\alpha} \| b_i^* \|^2 \\
\text{And for } k \neq i, i-1, \quad b_k^* = b_k
\end{align*}
\]

\[
\phi(B^t) = \prod_{i=1}^{n} \det(B_i B_i^t) \\
= \prod_{i=1}^{n} \det(B_i B_i^t) \\
= \prod_{i=1}^{n} \| b_i^* \|^2 \\
= \| b_1^* \|^2 \ldots \| b_n^* \|^2 \\
\| b_{i-1}^* \|^2 = \alpha \| b_{i-1}^* \|^2 \quad \text{and} \quad \| b_i^* \|^2 = \frac{1}{\alpha} \| b_i^* \|^2 \\
= \| b_1^* \|^2 \ldots \| b_n^* \|^2 \left( \alpha^{n-(i-2)} \right) \left( \frac{1}{\alpha} \right)^{n-(i-1)} \\
= \alpha \phi(B) \leq \frac{3}{4} \phi(B)
\]
Since the initial basis is rational, \( \exists \mathcal{g} \in \mathbb{N} \) each \( b_i \) initially has entries in \( \frac{1}{\mathcal{g}} \mathbb{Z} \)

\[ \Rightarrow \mathcal{L} \subseteq \frac{1}{\mathcal{g}} \mathbb{Z}^n = \left\{ \frac{1}{\mathcal{g}} x : x \in \mathbb{Z}^n \right\} \]

\[ \left( \frac{1}{\mathcal{g}} \right)^{2n^2} \]

is a lower bound on \( \phi(B) \) for any basis \( B \) of the lattice.

Initial basis \( \phi(B_{\text{initial}}) \leq (2\sqrt{nM})^{n^2} \)

\[ \Rightarrow \text{# steps } k \text{ is bounded by:} \]

\[ \left( \frac{3}{4} \right)^k (2\sqrt{nM})^{n^2} \geq \left( \frac{1}{\mathcal{g}} \right)^{2n^2} \]

\[ \Rightarrow (2\sqrt{nM} \mathcal{g}^2)^{n^2} \geq \left( \frac{4}{3} \right)^k \]

\[ \Rightarrow n^2 \log (2\sqrt{nM} \mathcal{g}^2) \geq k \log \frac{4}{3} \]

\[ k \leq n^2 \log (2\sqrt{nM} \mathcal{g}^2) \]
\[ \log \frac{1}{3} \]

\[ \text{# steps} \leq \sigma \left( n^2 \log \left( \frac{25nM}{\sigma^2} \right) \right) \]

\[ \text{poly} (n, \log M, \log \sigma) \]

\[ \det (B_i^T B_i) \]

\[ B_i = \begin{bmatrix} 1 & \ldots & 1 \\ b_i & \ldots & b_i \end{bmatrix} \]

\[ \det (X Y) \]

\[ \det (x_{ij}) \]

\[ \text{Cauchy-Binet} \quad S \subseteq \{1, \ldots, n\}^2 \quad |S| = i \]

\[ \det (B_i^T B_i) = \sum_{S} \det(A_S) \det(A_S^T) \]

where \( A_S \) is the \( i \times i \) square submatrix of \( B_i \) obtained by selecting the rows corresponding to \( S \).
\[ \sum_{i=1}^{\eta} \binom{M}{i} \binom{M}{\eta} \leq (1 + 5M)^{\eta} \]

**Integer Programming**

\[ \max \{ \langle c, x \rangle : x \in P, \sum x = \gamma \} \]

(binary search on \( \text{OPT} \))

Feasibility: \( Q \cap \mathbb{Z}^n = \emptyset ? \)

Polyhedron

Lenska:

Fix \( d \in \mathbb{N} \). Consider all polytopes in \( \mathbb{R}^d \).

Then feasibility question can be solved in time \( \text{poly}(\text{size}(A, b)) \)

\[ A \leq b \]

20(d^3)

\[ \text{poly}(d, \text{size}(A, b)) \]

20(d log d)

\[ \text{poly}(d, \ldots) \]

Q: Given a compact, convex set \( C \subset \mathbb{R}^d \)
equipped with a separation oracles.

\[ C \cap \mathbb{Z}^d = \emptyset ? \]
Def: An ellipsoid is the image of the unit ball $B(0,1)$ under an invertible affine transformation.

$E = \left\{ A \cdot x + b : \|x\|_2 \leq 1 \right\}$

$A: \mathbb{R}^d \to \mathbb{R}^d$ invertible

$E = \left\{ y \in \mathbb{R}^d : (y - c)^T B (y - c) \leq 1 \right\}$

for some PD matrix $B \in \mathbb{R}^{d \times d}$

and center $c \in \mathbb{R}^d$

The eigenvectors of $B$ give the principal axes of $E$.

Thus: Let $C \subseteq \mathbb{R}^d$ be a full-dim. compact convex set.

Then $E$ an ellipsoid $E \subseteq \mathbb{R}^d$ s.t.

$\frac{1}{2} E \subseteq C \subseteq E$

$\exists \frac{1}{2}(y - c) + c : y \in E \exists$
Theorem (Weaker version of Kannan–John theorem)

Let $C \subseteq \mathbb{R}^d$ be a full-dim, compact convex set. 

$B(\varepsilon, r) \subseteq C \subseteq B(0, R)$

Then there exists an algorithm that returns an ellipsoid $E$ s.t.

$\frac{1}{2^{\frac{1}{2d}}} \leq \frac{E}{E'} \leq E$

and the algorithm has run-time (including separation oracle calls)

bounded by $\text{poly}(d, \log \frac{R}{\varepsilon})$

$\text{vol}(E) \leq e^{-\frac{1}{2d}} \text{vol}(E)$

$\text{vol}(E') \leq e^{-\frac{1}{5d^2}} \text{vol}(E)$
Query all 2d pts of the form
\[ C + \frac{1}{d+1} (V_i - C) \]
i to a principal axis direction.

**Case 1:** At least one of these pts is not in \( C \), and we receive a separating hyperplane. Then we construct a new ellipsoid that contains \( C \) and has a reduction in volume by a factor of \( e^{-\frac{1}{2d+2}} \).

**Case 2:** All these 2d pts are in \( C \).
After poly \((d, \log \left( \frac{R}{\varepsilon} \right))\) we are in Case 2.