The LLL Algorithm

Liam Hardiman

May 29, 2019

Two lattices

- Recall that the **lattice**, L, generated by the linearly independent vectors $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$ is the $\mathbb{Z}$-span of these vectors:

$$L = \{c_1 x_1 + c_2 x_2 + \cdots + c_n x_n : c_i \in \mathbb{Z}, 1 \leq i \leq n\}.$$

Two lattices

- Recall that the **lattice**, L, generated by the linearly independent vectors $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$ is the $\mathbb{Z}$-span of these vectors:

$$L = \{c_1x_1 + c_2x_2 + \cdots + c_nx_n : c_i \in \mathbb{Z}, 1 \leq i \leq n\}.$$

- Consider the lattices, L and M, generated by the rows of the matrices X and Y, respectively.

$$X = \begin{bmatrix} -168 & 602 & 58 \\ 157 & -564 & -57 \\ 594 & -2134 & -219 \end{bmatrix}, \quad Y = \begin{bmatrix} -6 & 6 & -4 \\ 9 & 4 & 1 \\ -1 & 8 & 6 \end{bmatrix}.$$

Two lattices

- Each row of X is an integer linear combination of the rows of Y, so $L \subseteq M$:

Two lattices

- Each row of X is an integer linear combination of the rows of Y, so $L \subseteq M$:

$$\begin{bmatrix} -168 \\ 602 \\ 58 \end{bmatrix}^T = 14 \begin{bmatrix} 4 \\ 2 \\ -9 \end{bmatrix}^T + 50 \begin{bmatrix} -1 \\ 8 \\ 6 \end{bmatrix}^T - 29 \begin{bmatrix} 6 \\ -6 \\ 4 \end{bmatrix}^T,$$

$$\begin{bmatrix} 157 \\ -564 \\ -57 \end{bmatrix}^T = -13 \begin{bmatrix} 4 \\ 2 \\ -9 \end{bmatrix}^T - 47 \begin{bmatrix} -1 \\ 8 \\ 6 \end{bmatrix}^T + 26 \begin{bmatrix} 6 \\ -6 \\ 4 \end{bmatrix}^T,$$

$$\begin{bmatrix} 594 \\ -2134 \\ -219 \end{bmatrix} = -49 \begin{bmatrix} 4 \\ 2 \\ -9 \end{bmatrix}^T - 178 \begin{bmatrix} -1 \\ 8 \\ 6 \end{bmatrix} + 102 \begin{bmatrix} 6 \\ -6 \\ 4 \end{bmatrix}^T.$$

Two lattices

- In particular, we have the matrix equation

$$UY = X,$$

$$\begin{bmatrix} 14 & 50 & -29 \\ -13 & -47 & 27 \\ -49 & -178 & 102 \end{bmatrix} \begin{bmatrix} 4 & 2 & -9 \\ -1 & 8 & -6 \\ 6 & -6 & 4 \end{bmatrix} = \begin{bmatrix} -168 & 602 & 58 \\ 157 & -564 & -57 \\ 594 & -2134 & -219 \end{bmatrix}.$$

Two lattices

- In particular, we have the matrix equation

$$UY = X,$$

$$\begin{bmatrix} 14 & 50 & -29 \\ -13 & -47 & 27 \\ -49 & -178 & 102 \end{bmatrix} \begin{bmatrix} 4 & 2 & -9 \\ -1 & 8 & -6 \\ 6 & -6 & 4 \end{bmatrix} = \begin{bmatrix} -168 & 602 & 58 \\ 157 & -564 & -57 \\ 594 & -2134 & -219 \end{bmatrix}.$$

- $\det U = -1$, so U^{-1} is an integer matrix as well. This gives us another matrix equation, $Y = U^{-1}X$.

Two lattices

- In particular, we have the matrix equation

$$UY = X,$$

$$\begin{bmatrix} 14 & 50 & -29 \\ -13 & -47 & 27 \\ -49 & -178 & 102 \end{bmatrix} \begin{bmatrix} 4 & 2 & -9 \\ -1 & 8 & -6 \\ 6 & -6 & 4 \end{bmatrix} = \begin{bmatrix} -168 & 602 & 58 \\ 157 & -564 & -57 \\ 594 & -2134 & -219 \end{bmatrix}.$$

- $\det U = -1$, so U^{-1} is an integer matrix as well. This gives us another matrix equation, $Y = U^{-1}X$.
- Since the entries of U^{-1} are integers, this equation expresses the rows of Y as integer linear combinations of the rows of X, so $M \subseteq L$.

Two lattices

- Even though the rows of X and Y generate the same lattice, something about the Y-basis "feels" nicer.

$$X = \begin{bmatrix} -168 & 602 & 58 \\ 157 & -564 & -57 \\ 594 & -2134 & -219 \end{bmatrix}, \quad Y = \begin{bmatrix} -6 & 6 & -4 \\ 9 & 4 & 1 \\ -1 & 8 & 6 \end{bmatrix}.$$

Two lattices

- Even though the rows of X and Y generate the same lattice, something about the Y-basis "feels" nicer.

$$X = \begin{bmatrix} -168 & 602 & 58 \\ 157 & -564 & -57 \\ 594 & -2134 & -219 \end{bmatrix}, \quad Y = \begin{bmatrix} -6 & 6 & -4 \\ 9 & 4 & 1 \\ -1 & 8 & 6 \end{bmatrix}.$$

- Two qualities that make a basis desirable are:
    - Length: how long are the basis vectors?
    - Orthogonality: are the basis vectors nearly orthogonal to each other?

What makes a basis "nice"?

- Suppose $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$ are pairwise orthogonal.

What makes a basis "nice"?

- Suppose $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$ are pairwise orthogonal.
- If $x = c_1x_1 + c_2x_2 + \cdots + c_nx_n$, $c_i \in \mathbb{Z}$, is in the lattice L generated by $x_1, \ldots, x_n$ then

$$|x|^2 = c_1^2|x_1|^2 + c_2^2|x_2|^2 + \cdots + c_n^2|x_n|^2.$$

What makes a basis "nice"?

- Suppose $x_1, x_2, \ldots, x_n \in \mathbb{R}^n$ are pairwise orthogonal.
- If $x = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$, $c_i \in \mathbb{Z}$, is in the lattice L generated by $x_1, \ldots, x_n$ then

$$|x|^2 = c_1^2 |x_1|^2 + c_2^2 |x_2|^2 + \cdots + c_n^2 |x_n|^2.$$

- This completely solves the shortest vector problem (SVP) since

$$\arg\min_{x \in L} |x| = \arg\min_{x \in \{\pm x_1, \pm x_2, \ldots, \pm x_n\}} |x|.$$

What makes a basis "nice"?

- Say we want to find a vector in L that is closest to

$$x = t_1 x_1 + t_2 x_2 + \cdots + t_n x_n,$$

where the t_i are *real* numbers.

What makes a basis “nice”?

- Say we want to find a vector in L that is closest to

$$x = t_1 x_1 + t_2 x_2 + \cdots + t_n x_n,$$

where the t_i are *real* numbers.

- If $y = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$, $c_i \in \mathbb{Z}$, is any vector in L then by the orthogonality of the x_i we have

$$|x - y|^2 = (t_1 - c_1)^2 |x_1|^2 + (t_2 - c_2)^2 |x_2|^2 + \cdots + (t_n - c_n)^2 |x_n|^2.$$

What makes a basis "nice"?

- Say we want to find a vector in L that is closest to

$$x = t_1x_1 + t_2x_2 + \cdots + t_nx_n,$$

  where the t_i are *real* numbers.
- If $y = c_1x_1 + c_2x_2 + \cdots + c_nx_n$, $c_i \in \mathbb{Z}$, is any vector in L then by the orthogonality of the x_i we have

$$|x-y|^2 = (t_1-c_1)^2|x_1|^2+(t_2-c_2)^2|x_2|^2+\cdots+(t_n-c_n)^2|x_n|^2.$$

- If we take c_i to be the closest integer to t_i then we solve the closest vector problem (CVP).

Measuring orthogonality

Definition

Let $x_1, \ldots, x_n$ be a basis for the lattice $L \subset \mathbb{R}^n$. We define the determinant of L, $\det L$ to be the volume of the n-dimensional parallelepiped with sides defined by $x_1, \ldots, x_n$:

$$\det L = |\det X|,$$

where the rows of X are the basis vectors $x_1, \ldots, x_n$.

Measuring orthogonality

Definition

Let $x_1, \ldots, x_n$ be a basis for the lattice $L \subset \mathbb{R}^n$. We define the determinant of L, $\det L$ to be the volume of the n-dimensional parallelepiped with sides defined by $x_1, \ldots, x_n$:

$$\det L = |\det X|,$$

where the rows of X are the basis vectors $x_1, \ldots, x_n$.

Theorem

The determinant of L is independent of basis.

Measuring Orthogonality

- The volume of the parallelepiped defined by $x_1, \ldots, x_n$ is maximized when when the basis vectors are pairwise orthogonal to one another.

Measuring Orthogonality

- The volume of the parallelepiped defined by $x_1, \ldots, x_n$ is maximized when when the basis vectors are pairwise orthogonal to one another.

Theorem (Hadamard's Inequality)

Let $x_1, \ldots, x_n$ be a basis for the lattice $L \subset \mathbb{R}^n$. Then

$$\det L \leq |x_1||x_2| \cdots |x_n|.$$

Measuring Orthogonality

- The volume of the parallelepiped defined by $x_1, \ldots, x_n$ is maximized when when the basis vectors are pairwise orthogonal to one another.

Theorem (Hadamard's Inequality)

Let $x_1, \ldots, x_n$ be a basis for the lattice $L \subset \mathbb{R}^n$. Then

$$\det L \leq |x_1||x_2| \cdots |x_n|.$$

- If the basis vectors are closer to being orthogonal, then Hadamard's inequality is closer to an equality.

Gram-Schmidt

Definition

Let $x_1, \ldots, x_m \in \mathbb{R}^n$ be a basis for a nonzero subspace, H. The **Gram-Schmidt process** produces an orthogonal basis for H:

$$\begin{aligned} x_1^* &= x_1 \\ x_2^* &= x_2 - \mu_{2,1} x_1^* \\ x_3^* &= x_3 - \mu_{3,1} x_1^* - \mu_{3,2} x_2^* \\ &\vdots \\ x_m^* &= x_m - \mu_{m,1} x_1^* - \cdots - \mu_{m,m-1} x_{m-1}^*, \end{aligned}$$

where $\mu_{i,j} = \frac{x_i \cdot x_j^*}{x_j^* \cdot x_j^*}$. We call $\{x_1^*, \ldots, x_m^*\}$ the **Gram-Schmidt orthogonalization (GSO)** of $\{x_1, \ldots, x_m\}$.

Gram-Schmidt

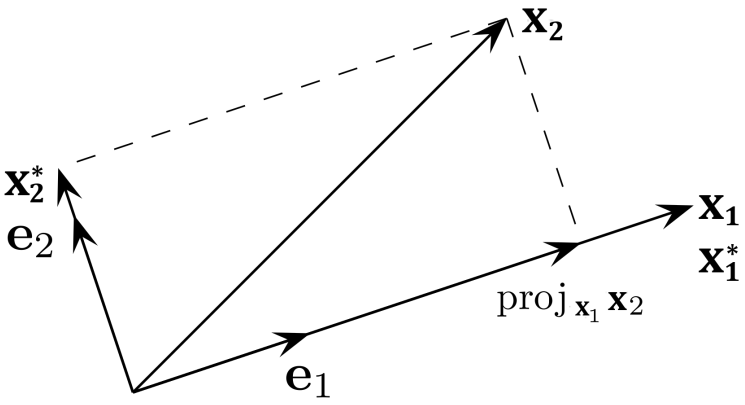

Figure: The first step of the Gram-Schmidt process. Image modified from `https://en.wikipedia.org/wiki/Gram-Schmidt_process`

Gram-Schmidt

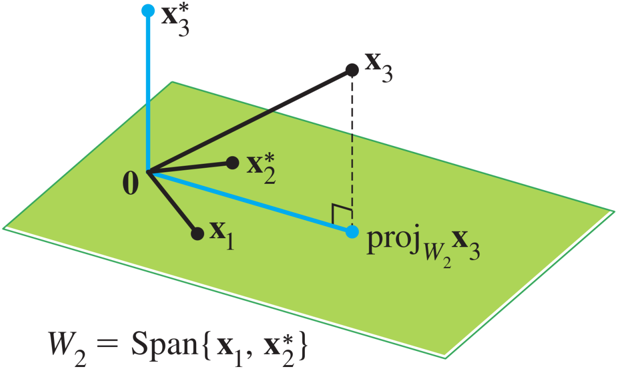

Figure: The second step of the Gram-Schmidt process. Image modified from D. Lay, S. Lay, and J. McDonald. *Linear Algebra and its Applications*. Fifth Edition. 2016.

A useful estimate

Proposition

Let $x_1, x_2, \ldots, x_n$ *be a basis for the lattice* $L \subset \mathbb{R}^n$ *and let* $x_1^*, x_2^*, \ldots, x_n^*$ *be its Gram-Schmidt orthogonalization. For any nonzero* $y \in L$ *we have*

$$|y| \geq \min\{|x_1^*|, |x_2^*|, \ldots, |x_n^*|\}.$$

That is, any nonzero lattice vector is at least as long as the shortest vector in the Gram-Schmidt orthogonalization.

A useful estimate

Proof

- We can write

$$y = \sum_{i=1}^{n} c_i x_i, \quad c_i \in \mathbb{Z}.$$

A useful estimate

Proof

- We can write
$$y = \sum_{i=1}^{n} c_i x_i, \quad c_i \in \mathbb{Z}.$$
- Since $y \neq 0$, at lease one c_i is nonzero. Let k be the largest index with $c_k \neq 0$.

A useful estimate

Proof

- We can write
$$y = \sum_{i=1}^{n} c_i x_i, \quad c_i \in \mathbb{Z}.$$
- Since $y \neq 0$, at lease one c_i is nonzero. Let k be the largest index with $c_k \neq 0$.

$$y = \sum_{i=1}^{k} \sum_{j=1}^{i} c_i \mu_{ij} x_j^* = \sum_{j=1}^{k} \left(\sum_{i=j}^{k} c_i \mu_{ij} \right) x_j^*$$
$$= c_k x_k^* + \sum_{j=1}^{k-1} \nu_j x_j^*,$$

for some real ν_j.

A useful estimate

Proof contd...

- Take the norm-squared on both sides.

$$
\begin{aligned}
|y|^2 &= \left| c_k x_k^* + \sum_{j=1}^{k-1} \nu_j x_j^* \right|^2 \\
&= c_k^2 |x_k^*|^2 + \sum_{j=1}^{k-1} \nu_j^2 |x_j^*|^2 \\
&\geq |x_k^*|^2 \\
&\geq \min\{|x_1^*|^2, |x_2^*|^2, \ldots, |x_n^*|^2\}.
\end{aligned}
$$

□

A useful equality

Proposition

If $x_1, \ldots, x_n$ is a basis for the lattice $L \subset \mathbb{R}^n$ and $x_1^*, \ldots, x_n^*$ is its GSO then

$$\det L = \prod_{i=1}^{n} |x_i^*|.$$

Proof

- We have that $\det L = \det X$, where the rows of X are the basis vectors $x_1, \ldots, x_n$.

A useful equality

Proof contd...

- By the definition of the GSO we have $X = MX^*$ where the rows of X^* are the vectors $x_1^*, \ldots, x_n^*$ and M consists of the GSO coefficients:

$$M = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \mu_{2,1} & 1 & 0 & \cdots & 0 & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_{n,1} & \mu_{n,2} & \mu_{n3} & \cdots & \mu_{n,n-1} & 1 \end{bmatrix}.$$

A useful equality

Proof contd...

- By the definition of the GSO we have $X = MX^*$ where the rows of X^* are the vectors $x_1^*, \ldots, x_n^*$ and M consists of the GSO coefficients:

$$M = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ \mu_{2,1} & 1 & 0 & \cdots & 0 & 0 \\ \mu_{3,1} & \mu_{3,2} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_{n,1} & \mu_{n,2} & \mu_{n3} & \cdots & \mu_{n,n-1} & 1 \end{bmatrix}.$$

- Since M has determinant 1 we have

$$\det L = |\det X| = |\det M||\det X^*| = \prod_{i=1}^{n} |x_i^*|.$$

Gram-Schmidt

- Given a basis $x_1, \ldots, x_n$ for a lattice $L \subset \mathbb{R}^n$, the GSO vectors $x_1^*, \ldots, x_n^*$ need not live in L since the coefficients $\frac{x_i \cdot x_j^*}{x_j^* \cdot x_j^*}$ need not be integers.

Gram-Schmidt

- Given a basis $x_1, \ldots, x_n$ for a lattice $L \subset \mathbb{R}^n$, the GSO vectors $x_1^*, \ldots, x_n^*$ need not live in L since the coefficients $\frac{x_i \cdot x_j^*}{x_j^* \cdot x_j^*}$ need not be integers.
- Can we salvage the Gram-Schmidt process and come up with a (nearly) orthogonal basis for L?

Basis reduction

Definition

Let α, $\frac{1}{4} < \alpha < 1$ be a real number. Let $x_1, \ldots, x_n$ be a basis for the lattice $L \subset \mathbb{R}^n$ and let $x_1^*, \ldots, x_n^*$ be its Gram-Schmidt orthogonalization. We say that the basis $x_1, \ldots, x_n$ is α-**reduced** if

1. (size condition) $|\mu_{ij}| \leq \frac{1}{2}$ for all $i \leq j < i \leq n$,
2. (Lovász condition) $|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2$ for $2 \leq i \leq n$.

Size condition

- $x_2 - \mu_{2,1}x_1$ is orthogonal to x_1, but might not be in the lattice spanned by $x_1, x_2, \ldots, x_n$.

Size condition

- $x_2 - \mu_{2,1}x_1$ is orthogonal to x_1, but might not be in the lattice spanned by $x_1, x_2, \ldots, x_n$.
- $x_2 - \lceil \mu_{2,1} \rfloor x_1$, where $\lceil x \rfloor$ is the integer closest to x (rounded down when $x = \frac{1}{2}$), *is* in the lattice and is nearly orthogonal to x_1.

Size condition

- $x_2 - \mu_{2,1}x_1$ is orthogonal to x_1, but might not be in the lattice spanned by $x_1, x_2, \ldots, x_n$.
- $x_2 - \lceil \mu_{2,1} \rfloor x_1$, where $\lceil x \rfloor$ is the integer closest to x (rounded down when $x = \frac{1}{2}$), *is* in the lattice and is nearly orthogonal to x_1.
- The size condition, $|\mu_{ij}| \leq \frac{1}{2}$, then says that $\lceil \mu_{ij} \rfloor = 0$: x_i is already nearly orthogonal to x_j.

Lovász condition

- Assuming the size condition is met, the Lovász condition, $|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2$ for all $i \geq 2$, says that x_i^* isn't isn't "too much" shorter than x_{i-1}.

Lovász condition

- Assuming the size condition is met, the Lovász condition, $|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2$ for all $i \geq 2$, says that x_i^* isn't isn't "too much" shorter than x_{i-1}.
- Rearranging gives $|x_i^* + \mu_{i,i-1}x_{i-1}^*|^2 \geq \alpha|x_{i-1}^*|^2$. This says

$$
\begin{aligned}
&|\text{Projection of } x_i \text{ onto Span}\{x_1, \ldots, x_{i-2}\}| \\
&\qquad \geq \alpha|\text{Projection of } x_{i-1} \text{ onto Span}\{x_1, \ldots, x_{i-2}\}|.
\end{aligned}
$$

Shortness

- Let $x_1, x_2, \ldots, x_n$ be an α-reduced basis and let $\beta = \frac{1}{\alpha - 1/4}$.

Shortness

- Let $x_1, x_2, \ldots, x_n$ be an α-reduced basis and let $\beta = \frac{1}{\alpha - 1/4}$.
- Combining the size and Lovász conditions gives

$$|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2 \geq \frac{1}{\beta}|x_{i-1}^*|^2.$$

Shortness

- Let $x_1, x_2, \ldots, x_n$ be an α-reduced basis and let $\beta = \frac{1}{\alpha - 1/4}$.
- Combining the size and Lovász conditions gives

$$|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2 \geq \frac{1}{\beta}|x_{i-1}^*|^2.$$

- Repeatedly applying this to $x_1^* = x_1$ gives

$$|x_1|^2 \leq \beta|x_2^*|^2 \leq \beta^2|x_3^*|^2 \leq \cdots \leq \beta^{n-1}|x_n^*|^2.$$

Shortness

- For any $2 \leq i \leq n$ we have $|x_i^*|^2 \geq \beta^{-(i-1)}|x_1|^2$.

Shortness

- For any $2 \leq i \leq n$ we have $|x_i^*|^2 \geq \beta^{-(i-1)}|x_1|^2$.
- If we let y be any nonzero vector in the lattice spanned by $x_1, \ldots, x_n$ we then have

$$|y| \geq \min\{|x_1^*|, \ldots, |x_n^*|\} \geq \beta^{-(n-1)/2}|x_1|.$$

Shortness

- For any $2 \leq i \leq n$ we have $|x_i^*|^2 \geq \beta^{-(i-1)}|x_1|^2$.
- If we let y be any nonzero vector in the lattice spanned by $x_1, \ldots, x_n$ we then have

$$|y| \geq \min\{|x_1^*|, \ldots, |x_n^*|\} \geq \beta^{-(n-1)/2}|x_1|.$$

- This gives a bound on the first vector in an α-reduced basis in terms of the shortest nonzero vector y in L:

$$|x_1| \leq \beta^{(n-1)/2}|y|.$$

Orthogonality

- If $x_1, \ldots, x_n$ is α-reduced, the Lovász condition gives us

$$|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2 \geq \frac{1}{\beta}|x_{i-1}^*|^2.$$

Orthogonality

- If $x_1, \ldots, x_n$ is α-reduced, the Lovász condition gives us

$$|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2 \geq \frac{1}{\beta}|x_{i-1}^*|^2.$$

- Repeated application gives $|x_j^*|^2 \leq \beta^{i-j}|x_i^*|^2$.

Orthogonality

- If $x_1, \ldots, x_n$ is α-reduced, the Lovász condition gives us

$$|x_i^*|^2 \geq (\alpha - \mu_{i,i-1}^2)|x_{i-1}^*|^2 \geq \frac{1}{\beta}|x_{i-1}^*|^2.$$

- Repeated application gives $|x_j^*|^2 \leq \beta^{i-j}|x_i^*|^2$.
- Writing x_i in terms of the GSO, $x_1^*, \ldots, x_n^*$ and applying the above inequality gives

$$|x_i|^2 \leq \beta^{i-1}|x_i^*|^2.$$

Orthogonality

- Multiplying this inequality by itself for $1 \leq i \leq n$ gives

$$\prod_{i=1}^{n} |x_i|^2 \leq \beta^{n(n-1)/2} \prod_{i=1}^{n} |x_i^*|^2 = \beta^{n(n-1)/2} (\det L)^2.$$

Orthogonality

- Multiplying this inequality by itself for $1 \leq i \leq n$ gives

$$\prod_{i=1}^{n} |x_i|^2 \leq \beta^{n(n-1)/2} \prod_{i=1}^{n} |x_i^*|^2 = \beta^{n(n-1)/2} (\det L)^2.$$

- Taking the square root and using Hadamard's inequality we have

$$\det L \leq \prod_{i=1}^{n} |x_i| \leq \beta^{n(n-1)/4} \det L.$$

Can we find a reduced basis?

- Reduced bases are nice. Their vectors are short and nearly orthogonal.

Can we find a reduced basis?

- Reduced bases are nice. Their vectors are short and nearly orthogonal.
- Does every lattice admit a reduced basis? If it does, can we compute it efficiently?

The LLL algorithm

Algorithm 1 The LLL Algorithm

Input: A basis $\{x_1, \ldots, x_n\}$ of the lattice $L \subset \mathbb{R}^n$ and a reduction parameter $\alpha \in (\frac{1}{4}, 1)$.

Output: An α-reduced basis $\{y_1, \ldots, y_n\}$ of the lattice L.

Copy $x_1, \ldots, x_n$ into $y_1, \ldots, y_n$.
Set $k \leftarrow 2$
while $k \leq n$ **do**
    for $j = k-1, k-2, \ldots, 2, 1$ **do**
        Set $y_k \leftarrow y_k - \lceil \mu_{k,j} \rfloor y_j$.
    if $|y_k^*|^2 \geq (\alpha - \mu_{k,k-1}^2)|y_{k-1}^*|^2$ **then**
        Set $k \leftarrow k+1$.
    else
        Swap y_{k-1} and y_k.
        Set $k \leftarrow \max(k-1, 2)$.
return $\{y_1, \ldots, y_n\}$.

Why swap?

- If we let L_k be the lattice spanned by $y_1, \ldots, y_k$, then swapping y_k and y_{k-1} changes the sublattices L_k and L_{k-1}.

Why swap?

- If we let L_k be the lattice spanned by $y_1, \ldots, y_k$, then swapping y_k and y_{k-1} changes the sublattices L_k and L_{k-1}.
- If the Lovász condition isn't met then

$$|y_k^*|^2 < (\alpha - \mu_{k,k-1}^2)|y_{k-1}^*|^2.$$

Why swap?

- If we let L_k be the lattice spanned by $y_1, \ldots, y_k$, then swapping y_k and y_{k-1} changes the sublattices L_k and L_{k-1}.
- If the Lovász condition isn't met then

$$|y_k^*|^2 < (\alpha - \mu_{k,k-1}^2)|y_{k-1}^*|^2.$$

- Since $\det L_{k-1} = |y_1^*| \cdots |y_{k-1}^*|$, replacing y_{k-1}^* with the shorter y_k^* will decrease $\det L_{k-1}$.

Why swap?

- If we let L_k be the lattice spanned by $y_1, \ldots, y_k$, then swapping y_k and y_{k-1} changes the sublattices L_k and L_{k-1}.
- If the Lovász condition isn't met then

$$|y_k^*|^2 < (\alpha - \mu_{k,k-1}^2)|y_{k-1}^*|^2.$$

- Since $\det L_{k-1} = |y_1^*| \cdots |y_{k-1}^*|$, replacing y_{k-1}^* with the shorter y_k^* will decrease $\det L_{k-1}$.
- The swapping step attempts to order the vectors y_i so that the determinants of the sublattices, $\det L_i$, are minimized.

An example

- Let $L \subset \mathbb{R}^n$ be the $\mathbb{Z}$-span of the rows of the matrix X:

$$X = \begin{bmatrix} 4 & 5 & 1 \\ 4 & 8 & 2 \\ 6 & 2 & 6 \end{bmatrix}.$$

An example

- Let $L \subset \mathbb{R}^n$ be the $\mathbb{Z}$-span of the rows of the matrix X:

$$X = \begin{bmatrix} 4 & 5 & 1 \\ 4 & 8 & 2 \\ 6 & 2 & 6 \end{bmatrix}.$$

- Running the LLL algorithm with $\alpha = \frac{3}{4}$ gives the reduced basis

$$Y = \begin{bmatrix} 0 & 3 & 1 \\ 4 & -1 & -1 \\ 2 & -3 & 5 \end{bmatrix}.$$

An example

- Let $L \subset \mathbb{R}^n$ be the $\mathbb{Z}$-span of the rows of the matrix X:

$$X = \begin{bmatrix} 4 & 5 & 1 \\ 4 & 8 & 2 \\ 6 & 2 & 6 \end{bmatrix}.$$

- Running the LLL algorithm with $\alpha = \frac{3}{4}$ gives the reduced basis

$$Y = \begin{bmatrix} 0 & 3 & 1 \\ 4 & -1 & -1 \\ 2 & -3 & 5 \end{bmatrix}.$$

- It's clear that the vectors in the Y basis are shorter: the longest row vector in Y is shorter than the shortest row vector in X!

An example

- Let $L \subset \mathbb{R}^n$ be the $\mathbb{Z}$-span of the rows of the matrix X:

$$X = \begin{bmatrix} 4 & 5 & 1 \\ 4 & 8 & 2 \\ 6 & 2 & 6 \end{bmatrix}.$$

- Running the LLL algorithm with $\alpha = \frac{3}{4}$ gives the reduced basis

$$Y = \begin{bmatrix} 0 & 3 & 1 \\ 4 & -1 & -1 \\ 2 & -3 & 5 \end{bmatrix}.$$

- It's clear that the vectors in the Y basis are shorter: the longest row vector in Y is shorter than the shortest row vector in X!
- The Y basis is more orthogonal than the X basis:

$$\det L = 76, \quad |x_1||x_2||x_3| \approx 518, \quad |y_1||y_2||y_3| \approx 83$$

Does the algorithm terminate?

- Because we swap the vectors in the y_i basis, it's not obvious that the LLL algorithm terminates.

Does the algorithm terminate?

- Because we swap the vectors in the y_i basis, it's not obvious that the LLL algorithm terminates.
- For simplicity, let's assume our basis vectors x_i have integer entries. Define the quantities

$$d_\ell = \prod_{i=1}^{\ell} |x_i^*|^2, \qquad D = \prod_{i=1}^{n} d_i = \prod_{i=1}^{n} |x_i^*|^{2(n+1-i)}.$$

Does the algorithm terminate?

- Because we swap the vectors in the y_i basis, it's not obvious that the LLL algorithm terminates.
- For simplicity, let's assume our basis vectors x_i have integer entries. Define the quantities

$$d_\ell = \prod_{i=1}^{\ell} |x_i^*|^2, \qquad D = \prod_{i=1}^{n} d_i = \prod_{i=1}^{n} |x_i^*|^{2(n+1-i)}.$$

- D changes when we swap x_k and x_{k-1}, so the only terms that will contribute to this change are $|x_k^*|^2$ and $|x_{k-1}^*|^2$.

Does the algorithm terminate?

- Because we swap the vectors in the y_i basis, it's not obvious that the LLL algorithm terminates.
- For simplicity, let's assume our basis vectors x_i have integer entries. Define the quantities

$$d_\ell = \prod_{i=1}^{\ell} |x_i^*|^2, \qquad D = \prod_{i=1}^{n} d_i = \prod_{i=1}^{n} |x_i^*|^{2(n+1-i)}.$$

- D changes when we swap x_k and x_{k-1}, so the only terms that will contribute to this change are $|x_k^*|^2$ and $|x_{k-1}^*|^2$.
- We swap when the Lovász condition isn't met, i.e. when

$$|x_k^*|^2 < (\alpha - \mu_{k,k-1}^2)|x_{k-1}^*|^2 \leq \alpha |x_{k-1}^*|^2.$$

Does the algorithm terminate?

- Swapping x_k and x_{k-1} changes only d_{k-1} and it changes by:

$$\begin{aligned} d_{k-1}^{new} &= |x_1^*|^2 \cdots |x_{k-2}^*|^2 \cdot |x_k^*|^2 \\ &= |x_1^*|^2 \cdots |x_{k-2}^*|^2 \cdot |x_{k-1}^*|^2 \cdot \frac{|x_k^*|^2}{|x_{k-1}^*|^2} \\ &= d_{k-1}^{old} \cdot \frac{|x_k^*|^2}{|x_{k-1}^*|^2} \leq \alpha \cdot d_{k-1}^{old}. \end{aligned}$$

- If we execute N swaps then D must be reduced by a factor of at least α^N, since each swap changes exactly one d_i, reducing it by a factor of α, and D is the product of all the d_i's.

Does the algorithm terminate?

- Swapping x_k and x_{k-1} changes only d_{k-1} and it changes by:

$$
\begin{aligned}
d_{k-1}^{new} &= |x_1^*|^2 \cdots |x_{k-2}^*|^2 \cdot |x_k^*|^2 \\
&= |x_1^*|^2 \cdots |x_{k-2}^*|^2 \cdot |x_{k-1}^*|^2 \cdot \frac{|x_k^*|^2}{|x_{k-1}^*|^2} \\
&= d_{k-1}^{old} \cdot \frac{|x_k^*|^2}{|x_{k-1}^*|^2} \leq \alpha \cdot d_{k-1}^{old}.
\end{aligned}
$$

- If we execute N swaps then D must be reduced by a factor of at least α^N, since each swap changes exactly one d_i, reducing it by a factor of α, and D is the product of all the d_i's.

Does the algorithm terminate?

- Since our lattice vectors live in $\mathbb{Z}^n$, each d_i is a positive integer, so
$$D = \prod_{i=1}^{n} d_i \geq 1.$$

Does the algorithm terminate?

- Since our lattice vectors live in $\mathbb{Z}^n$, each d_i is a positive integer, so

$$D = \prod_{i=1}^{n} d_i \geq 1.$$

- D is a positive integer bounded away from zero that decreases by a factor of at least α after each swap.

Does the algorithm terminate?

- Since our lattice vectors live in $\mathbb{Z}^n$, each d_i is a positive integer, so
$$D = \prod_{i=1}^{n} d_i \geq 1.$$
- D is a positive integer bounded away from zero that decreases by a factor of at least α after each swap.
- Such a positive integer can be multiplied by α only finitely many times before dropping below 1, so we must execute only finitely many swaps: the LLL algorithm terminates.

Runtime

- The LLL algorithm is only practical if it finishes execution in a reasonable amount of time.

Runtime

- The LLL algorithm is only practical if it finishes execution in a reasonable amount of time.
- Say the algorithm terminates after N swaps and let D_{init} be the value of D in the input basis. Since D decreases by a factor of α after each swap, we have that

$$1 \leq D_{final} \leq \alpha^N D_{init}.$$

Runtime

- The LLL algorithm is only practical if it finishes execution in a reasonable amount of time.
- Say the algorithm terminates after N swaps and let D_{init} be the value of D in the input basis. Since D decreases by a factor of α after each swap, we have that

$$1 \leq D_{final} \leq \alpha^N D_{init}.$$

- Taking logarithms and using $\log \alpha < 0$ gives

$$N = O(\log D_{init}).$$

Runtime

- How large is D_{init} in terms of the input basis?

Runtime

- How large is D_{init} in terms of the input basis?
- Since $|x_i^*| \leq |x_i|$ for all i by the Pythagorean theorem we have

$$
\begin{aligned}
D_{init} &= \prod_{i=1}^{n} |x_i^*|^{2(n+1-i)} \\
&\leq \prod_{i=1}^{n} |x_i|^{2(n+1-i)} \\
&\leq \big(\max_{1 \leq i \leq n} |x_i|\big)^{2(1+2+\cdots+n)} \\
&= B^{n^2+n},
\end{aligned}
$$

where B is the length of the longest vector in the input basis, $x_1, \ldots, x_n$.

Putting it all together

We have proved the following theorem.

Theorem (Lenstra, Lenstra, Lovász (1982))

Let $x_1, x_2, \ldots, x_n$ be a basis for the lattice $L \subset \mathbb{R}^n$ and let $\alpha \in (1/4, 1)$. There exists an α-reduced basis for L that can be computed in time

$$O(n^2 \log B),$$

where $B = \max_{1 \leq i \leq n} |x_i|$.

Small roots of polynomials mod M (D. Coppersmith '96)

- Suppose we know that $f(x) \in \mathbb{Z}[x]$ has a small root, x_0 modulo M and we want to find x_0: think $f(x) = x^e - c$ where M is an RSA modulus.

Small roots of polynomials mod M (D. Coppersmith '96)

- Suppose we know that $f(x) \in \mathbb{Z}[x]$ has a small root, x_0 modulo M and we want to find x_0: think $f(x) = x^e - c$ where M is an RSA modulus.
- We can use Newton's method to approximate the roots to $f(x)$, but these roots might not be roots mod M unless the coefficients of $f(x)$ are small.

Small roots of polynomials mod M (D. Coppersmith '96)

- Suppose we know that $f(x) \in \mathbb{Z}[x]$ has a small root, x_0 modulo M and we want to find x_0: think $f(x) = x^e - c$ where M is an RSA modulus.
- We can use Newton's method to approximate the roots to $f(x)$, but these roots might not be roots mod M unless the coefficients of $f(x)$ are small.
- Plan: build a polynomial $g(x) \in \mathbb{Z}[x]$ that has the same root x_0 modulo M as $f(x)$, but with coefficients small enough that $g(x_0) = 0$ as well.

From polynomials to lattices

- Suppose we know that $|x_0| < X$ for some integer X. Write $f(x) = a_0 + a_1 x + \cdots + a_d x^d$, $a_i \in \mathbb{Z}$. Consider the matrix

$$B = \begin{bmatrix} M & 0 & \cdots & 0 & 0 \\ 0 & MX & \cdots & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ 0 & 0 & \cdots & MX^{d-1} & 0 \\ a_0 & a_1 X & \cdots & a_{d-1}X^{d-1} & a_d X^d \end{bmatrix}.$$

From polynomials to lattices

- Suppose we know that $|x_0| < X$ for some integer X. Write $f(x) = a_0 + a_1 x + \cdots + a_d x^d$, $a_i \in \mathbb{Z}$. Consider the matrix

$$B = \begin{bmatrix} M & 0 & \cdots & 0 & 0 \\ 0 & MX & \cdots & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ 0 & 0 & \cdots & MX^{d-1} & 0 \\ a_0 & a_1 X & \cdots & a_{d-1}X^{d-1} & a_d X^d \end{bmatrix}.$$

- The rows of B are linearly independent and span a lattice $L \subset \mathbb{R}^{d+1}$.

From polynomials to lattices

- Suppose we know that $|x_0| < X$ for some integer X. Write $f(x) = a_0 + a_1 x + \cdots + a_d x^d$, $a_i \in \mathbb{Z}$. Consider the matrix

$$B = \begin{bmatrix} M & 0 & \cdots & 0 & 0 \\ 0 & MX & \cdots & 0 & 0 \\ \vdots & & & \vdots & \vdots \\ 0 & 0 & \cdots & MX^{d-1} & 0 \\ a_0 & a_1 X & \cdots & a_{d-1}X^{d-1} & a_d X^d \end{bmatrix}.$$

- The rows of B are linearly independent and span a lattice $L \subset \mathbb{R}^{d+1}$.
- Each row vector in L is of the form $(b_0, b_1 X, \ldots, b_d X^d)$, $b_i \in \mathbb{Z}$.

From polynomials to lattices

- Identify elements of L with polynomials in $\mathbb{Z}[x]$ by

$$(b_0, b_1X, \ldots, b_dX^d) \mapsto b_0 + b_1x + \cdots + b_dx^d.$$

From polynomials to lattices

- Identify elements of L with polynomials in $\mathbb{Z}[x]$ by

$$(b_0, b_1 X, \ldots, b_d X^d) \mapsto b_0 + b_1 x + \cdots + b_d x^d.$$

- Under this identification, each row of B corresponds to a polynomial with root x_0 modulo M. Consequently, every element in L corresponds to a polynomial with the same property.

How small should the coefficients be?

Theorem (N. Howgrave-Graham ('97))

Let b_F be a vector in $L \subset \mathbb{R}^d$ and let $F(x)$ be the corresponding polynomial in $\mathbb{Z}[x]$. If $|b_F| \leq M/\sqrt{d+1}$ then $F(x_0) = 0$.

How small should the coefficients be?

Theorem (N. Howgrave-Graham ('97))

Let b_F be a vector in $L \subset \mathbb{R}^d$ and let $F(x)$ be the corresponding polynomial in $\mathbb{Z}[x]$. If $|b_F| \leq M/\sqrt{d+1}$ then $F(x_0) = 0$.

- This tells us how small the coefficients of $F(x)$ need to be in order for x_0 to be a root of $F(x)$ mod M and $F(x_0) = 0$.

Applying the LLL algorithm

- The lattice L generated by B has determinant $M^d X^{d(d+1)/2}$.

Applying the LLL algorithm

- The lattice L generated by B has determinant $M^d X^{d(d+1)/2}$.
- Apply the LLL algorithm to the matrix B to obtain an α-reduced basis for the lattice L, $y_1, \ldots, y_d$.

Applying the LLL algorithm

- The lattice L generated by B has determinant $M^d X^{d(d+1)/2}$.
- Apply the LLL algorithm to the matrix B to obtain an α-reduced basis for the lattice L, $y_1, \ldots, y_d$.
- Recall that $|y_1|^2 \leq \beta^{i-1}|y_i^*|^2$ for all $1 \leq i \leq d+1$. Using this (and $\beta \leq 2$) we obtain the bound

$$|y_1| \leq 2^{d/4}(\det L)^{1/(d+1)} = 2^{d/4} M^{d/(d+1)} X^{d/2}.$$

Applying the LLL algorithm

- Howgrave-Graham's theorem tells us how small y_1 needs to be in order for it to correspond to a polynomial $g(x)$ such that $g(x_0) = 0$. This lets us solve for X to obtain:

$$\begin{aligned} & |y_1| < M/\sqrt{d+1} \\ \iff & 2^{d/4} M^{d/(d+1)} X^{d/2} < M/\sqrt{d+1} \\ \iff & X < 2^{-1/2}(d+1)^{-1/d} M^{2/d(d+1)}. \end{aligned}$$

Coppersmith's theorem

Theorem (D. Coppersmith ('96))

Let $f(x)$ be an integer polynomial with small root x_0 modulo M and $|x_0| < X$. If $X < 2^{-1/2}(d+1)^{-1/d} M^{2/d(d+1)}$ then there exists an algorithm that computes x_0 in time polynomial in the size of the coefficients of $f(x)$ and the degree of f.

Coppersmith's theorem

Theorem (D. Coppersmith ('96))

Let $f(x)$ be an integer polynomial with small root x_0 modulo M and $|x_0| < X$. If $X < 2^{-1/2}(d+1)^{-1/d}M^{2/d(d+1)}$ then there exists an algorithm that computes x_0 in time polynomial in the size of the coefficients of $f(x)$ and the degree of f.

- This gives an efficient attack on RSA implementations with small encryption exponents.