Algorithms for Lattice Basis Reduction

Curtis Bright

December 15, 2008

Abstract

This report contains an exposition of the theory behind the Lenstra-Lenstra-Lovász lattice basis reduction algorithm [2] and its precursors.

1 Introduction

The primary mathematical object studied in this report is the *lattice*. Given d linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{R}^n$, a lattice $L \subset \mathbb{R}^n$ generated by $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_d\}$ is defined as

$$L(B) = \left\{ \sum_{i=1}^{d} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}.$$

That is, the "integer span" of vectors in B (abbreviated $\sum_i \mathbb{Z}\mathbf{b}_i$). This report deals with the case when the $\mathbf{b}_i$ are integer vectors.

The set of vectors B is a *basis* of L. When $|B| > 1$ the lattice $L(B)$ has an infinite number of bases, but most are cumbersome to work with: the goal of LLL is to find nice or *reduced* bases. For example, the row vectors in the matrix

$$B = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix} = \begin{bmatrix} 109983 & 38030 & 97734 \\ 330030 & 114118 & 293274 \\ 277753 & 124767 & 173357 \end{bmatrix}$$

generate a lattice in $\mathbb{R}^3$. However, the row vectors in

$$B' = \begin{bmatrix} \mathbf{b}'_1 \\ \mathbf{b}'_2 \\ \mathbf{b}'_3 \end{bmatrix} = \begin{bmatrix} -15 & 6 & -42 \\ -47 & 25 & 11 \\ 4 & 65 & -1 \end{bmatrix}$$

define the same lattice, and in general is a much easier basis to work with.

The fact that B and B' generate the same lattice may be seen by the existence of the change-of-basis matrix

$$C = \begin{bmatrix} -2715 & -1358 & 1358 \\ -8147 & -4075 & 4075 \\ -5243 & -3904 & 3905 \end{bmatrix},$$

and it follows

$$B = CB'. \tag{1}$$

The entries in the change-of-basis from B to B' (and vice versa) must be integers: if $\mathbf{b} \in L(B)$, then $\mathbf{b} = \sum_i x_i \mathbf{b}_i$ for some $\mathbf{x} \in \mathbb{R}^d$. Using the change-of-basis matrix we get $\mathbf{b} = \sum_i \sum_j x_j c_{ij} \mathbf{b}'_i$, requiring $\sum_j x_j c_{ij} \in \mathbb{Z}$, which is only true when $c_{ij} \in \mathbb{Z}$ for arbitrary $\mathbf{x} \in \mathbb{Z}^d$.

Therefore, both C and C^{-1} must have integer entries: such matrices are known as *unimodular* and they have determinant ± 1. Then (1) yields

$$|\det(B)| = |\det(B')|,$$

so all bases of a lattice have the same absolute determinant, denoted $\mathrm{vol}(L)$ after the volume of the d-dimensional parallelotope formed by the $[0, 1)$-span of the basis vectors:

$$\mathrm{vol}(L) = |\det(B)| = \text{volume} \left\{ \sum_{i=1}^{d} x_i \mathbf{b}_i : x_i \in [0, 1) \right\}.$$

Although the basis used is mathematically irrelevant, shorter vectors are easier to work with. Taking this view, the best possible basis would have $\mathbf{b}_1$ as the shortest nonzero vector in the lattice[1] and in general $\mathbf{b}_i$ as the shortest nonzero vector such that $\mathbf{b}_1, \ldots, \mathbf{b}_i$ is linearly independent. Such a basis is called *minimal*.

2 Gram-Schmidt Orthogonalization

Given a basis $\mathbf{b}_1, \ldots, \mathbf{b}_d$ for a subspace of $\mathbb{R}^n$, the Gram-Schmidt process finds an orthogonal basis $\mathbf{b}^*_1, \ldots, \mathbf{b}^*_d$ of that subspace. Since a lattice is not a subspace, it cannot be directly used to find a new lattice basis, but will nevertheless be an important part of the algorithms we will see. The orthogonal basis is computed as follows:

$$\begin{aligned}
\mathbf{b}^*_1 &= \mathbf{b}_1 \\
\mathbf{b}^*_2 &= \mathbf{b}_2 - \mathrm{proj}_{\mathbf{b}^*_1} \mathbf{b}_2 \\
\mathbf{b}^*_3 &= \mathbf{b}_3 - \mathrm{proj}_{\mathbf{b}^*_1} \mathbf{b}_3 - \mathrm{proj}_{\mathbf{b}^*_2} \mathbf{b}_3 \\
&\vdots \\
\mathbf{b}^*_d &= \mathbf{b}_d - \sum_{j=1}^{d-1} \mathrm{proj}_{\mathbf{b}^*_j} \mathbf{b}_d
\end{aligned}$$

Intuitively, $\mathbf{b}^*_i$ is the component of $\mathbf{b}_i$ which is orthogonal to $\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}$, i.e.,

$$\mathbf{b}^*_i = \mathrm{proj}_{\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^\perp} \mathbf{b}_i.$$

[1] There will always be at least two nonzero vectors of shortest norm, but ties can be broken arbitrarily.

Let $\mu_{i,j}$ be the coefficient used in $\text{proj}_{\mathbf{b}_j^*} \mathbf{b}_i$, i.e.,

$$\mu_{i,j} = \frac{\mathbf{b}_i \cdot \mathbf{b}_j^*}{\mathbf{b}_j^* \cdot \mathbf{b}_j^*} = \frac{\mathbf{b}_i \cdot \mathbf{b}_j^*}{\left\|\mathbf{b}_j^*\right\|^2}.$$

In matrix form, Gram-Schmidt reads:

$$\begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_d \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ \mu_{2,1} & 1 & & & \\ \mu_{3,1} & \mu_{3,2} & 1 & & \\ \vdots & & & \ddots & \\ \mu_{d,1} & \mu_{d,2} & \cdots & \mu_{d,d-1} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_1^* \\ \vdots \\ \mathbf{b}_d^* \end{bmatrix}$$

Since the $\mathbf{b}_i^*$ are orthogonal, taking the determinant:

$$\begin{aligned} \text{vol}(L(B)) &= \prod_{i=1}^{d} \|\mathbf{b}_i^*\| \leq \prod_{i=1}^{d} \|\mathbf{b}_i\| \\ \text{since} \qquad \left\|\mathbf{b}_i^*\right\|^2 &\leq \left\|\mathbf{b}_i^*\right\|^2 + \left\|\textstyle\sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*\right\|^2 \\ &= \left\|\mathbf{b}_i^* + \textstyle\sum_{j=1}^{i-1} \mu_{i,j}\mathbf{b}_j^*\right\|^2 \qquad (\mathbf{b}_i^* \cdot \mathbf{b}_j^* = 0) \\ &= \left\|\mathbf{b}_i\right\|^2 \end{aligned}$$

Intuitively, $\prod_{i=1}^{d} \|\mathbf{b}_i\|$ measures the "nonorthogonality" in a basis. If it is approximately $\text{vol}(L(B))$ then $\mathbf{b}_1, \ldots, \mathbf{b}_d$ are almost orthogonal. The reductions we'll see will bound the permitted amount of nonorthogonality in a basis.

3 Lagrange's 2D Algorithm

To motivate the reductions in general dimension, we first describe an algorithm which finds a minimal basis in two dimensions. It was known to Lagrange in 1773, though it is also sometimes called Gauss' Algorithm. It is similar in style to Euclid's famous gcd algorithm: the vector norms are continually decreased by subtracting multiples of one vector from the other.

Algorithm 1 LagrangeReduce($\mathbf{b}_1$, $\mathbf{b}_2$)

Input: A basis $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^2$ for a lattice L.
Output: A minimal basis $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^2$ of L.

1. **repeat**
2. **if** $\|\mathbf{b}_1\| > \|\mathbf{b}_2\|$ **then**
3. swap $\mathbf{b}_1$ and $\mathbf{b}_2$
4. **end if**
5. $\mu_{2,1} := (\mathbf{b}_1 \cdot \mathbf{b}_2) \,/\, \|\mathbf{b}_1\|^2$
6. $\mathbf{b}_2 := \mathbf{b}_2 - \lfloor \mu_{2,1} \rceil \, \mathbf{b}_1$
7. **until** $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|$

After line 6 we could perform the update $\mu_{2,1} := \mu_{2,1} - \lfloor \mu_{2,1} \rceil$, which shows at the end of the loop we have $|\mu_{2,1}| \leq \frac{1}{2}$. Along with $\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|$, this is enough to show $\mathbf{b}_1, \mathbf{b}_2$ is a minimal basis. Let $\mathbf{b} = \alpha\mathbf{b}_1 + \beta\mathbf{b}_2$ be an arbitrary nonzero element of L.

First, we will show $\|\mathbf{b}_1\| \leq \|\mathbf{b}\|$, i.e., $\mathbf{b}_1$ is the smallest nonzero vector.

$$\begin{aligned}
\|\mathbf{b}\|^2 &= \|\alpha\mathbf{b}_1 + \beta\mathbf{b}_2\|^2 \\
&= \alpha^2 \|\mathbf{b}_1\|^2 + 2\alpha\beta\,(\mathbf{b}_1 \cdot \mathbf{b}_2) + \beta^2 \|\mathbf{b}_2\|^2 \\
&\geq \alpha^2 \|\mathbf{b}_1\|^2 - \alpha\beta \|\mathbf{b}_1\|^2 + \beta^2 \|\mathbf{b}_2\|^2 && (-\tfrac{1}{2} \leq \mu_{2,1}) \\
&\geq \left(\alpha^2 - \alpha\beta + \beta^2\right) \|\mathbf{b}_1\|^2 && (\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\|) \\
&\geq \|\mathbf{b}_1\|^2 && (\alpha \neq 0 \text{ or } \beta \neq 0)
\end{aligned}$$

Next, we will show $\|\mathbf{b}_2\| \leq \|\mathbf{b}\|$ when $\beta \neq 0$, i.e., $\mathbf{b}_2$ is the smallest nonzero vector not a multiple of $\mathbf{b}_1$.

$$\begin{aligned}
\|\mathbf{b}\|^2 &= \|\alpha\mathbf{b}_1 + \beta\mathbf{b}_2\|^2 \\
&= \alpha^2 \|\mathbf{b}_1\|^2 + 2\alpha\beta\,(\mathbf{b}_1 \cdot \mathbf{b}_2) + \beta^2 \|\mathbf{b}_2\|^2 + \left(\beta^2 \|\mathbf{b}_1\|^2 - \beta^2 \|\mathbf{b}_1\|^2\right) \\
&= \beta^2 \left(\|\mathbf{b}_2\|^2 - \|\mathbf{b}_1\|^2\right) + \left(\alpha^2 + \beta^2\right) \|\mathbf{b}_1\|^2 + 2\alpha\beta\,(\mathbf{b}_1 \cdot \mathbf{b}_2) \\
&\geq \beta^2 \left(\|\mathbf{b}_2\|^2 - \|\mathbf{b}_1\|^2\right) + \left(\alpha^2 - \alpha\beta + \beta^2\right) \|\mathbf{b}_1\|^2 \\
&\geq \beta^2 \left(\|\mathbf{b}_2\|^2 - \|\mathbf{b}_1\|^2\right) + \|\mathbf{b}_1\|^2 + \left(\|\mathbf{b}_2\|^2 - \|\mathbf{b}_2\|^2\right) \\
&= \left(\beta^2 - 1\right) \left(\|\mathbf{b}_2\|^2 - \|\mathbf{b}_1\|^2\right) + \|\mathbf{b}_2\|^2 \\
&\geq \|\mathbf{b}_2\|^2 && (\beta \neq 0)
\end{aligned}$$

Also, $\min\{\|\mathbf{b}_1\|, \|\mathbf{b}_2\|\}$ strictly decreases on every iteration except the last, and there are only finitely many lattice vectors shorter than any constant, so the algorithm must terminate. See the attached figure for an example of the algorithm running on the lattice with basis

$$B = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} 1 & \sqrt{3} \\ 1 & -\sqrt{3} \end{bmatrix}. \tag{2}$$

Lastly, we give an upper bound on $\|\mathbf{b}_1\| \|\mathbf{b}_2\|$ in terms of $\mathrm{vol}(L(B))$:

$$\begin{aligned}
\|\mathbf{b}_2\|^2 &= \|\mathbf{b}_2^* + \mu_{2,1}\mathbf{b}_1^*\|^2 \\
&= \|\mathbf{b}_2^*\|^2 + 2\mu_{2,1}\,(\mathbf{b}_2^* \cdot \mathbf{b}_1^*) + \mu_{2,1}^2 \|\mathbf{b}_1^*\|^2 \\
&= \|\mathbf{b}_2^*\|^2 + \mu_{2,1}^2 \|\mathbf{b}_1\|^2 \\
&\leq \|\mathbf{b}_2^*\|^2 + \tfrac{1}{4} \|\mathbf{b}_2\|^2 \\
\tfrac{3}{4} \|\mathbf{b}_2\|^2 &\leq \|\mathbf{b}_2^*\|^2 \\
\|\mathbf{b}_2\| &\leq \tfrac{2}{\sqrt{3}} \|\mathbf{b}_2^*\| \\
\|\mathbf{b}_1\| \|\mathbf{b}_2\| &\leq \tfrac{2}{\sqrt{3}} \,\mathrm{vol}(L(B))
\end{aligned}$$

Note that the basis (2) has $\|\mathbf{b}_1\| \, \|\mathbf{b}_2\| = \frac{2}{\sqrt{3}} \operatorname{vol}(L(B))$, showing that $\gamma_2 = \frac{2}{\sqrt{3}}$ is the best possible upper bound in dimension 2.

4 Hermite's Algorithm

In the 1840s Hermite described two slightly different lattice reduction algorithms in letters to Jacobi. Here we discuss his second algorithm, which is a generalization of Lagrange's Algorithm to n dimensions.

Algorithm 2 HERMITEREDUCE$(d, \mathbf{b}_1, \ldots, \mathbf{b}_d)$

Input: A basis $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{R}^n$ for a lattice L.
Output: A Hermite-reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{R}^n$ of L.

repeat
  if $\|\mathbf{b}_1\| > \|\mathbf{b}_i\|$ for some $i = 2, \ldots, d$ **then**
    swap $\mathbf{b}_1$ and $\mathbf{b}_i$, where $\mathbf{b}_i$ has minimal norm
  end if
  for $i = 2$ to d **do**
    $\mu_{i,1} := (\mathbf{b}_1 \cdot \mathbf{b}_i) \,/\, \|\mathbf{b}_1\|^2$
  end for
  if $d > 2$ **then**
    HERMITEREDUCE$(d-1, \mathbf{b}_2 - \mu_{2,1}\mathbf{b}_1, \ldots, \mathbf{b}_d - \mu_{d,1}\mathbf{b}_d)$
    apply reduction operations from above to $(\mathbf{b}_2, \ldots, \mathbf{b}_d)$
  end if
  for $i = 2$ to d **do**
    $\mu_{i,1} := (\mathbf{b}_1 \cdot \mathbf{b}_i) \,/\, \|\mathbf{b}_1\|^2$
    $\mathbf{b}_i := \mathbf{b}_i - \lfloor \mu_{i,1} \rceil \, \mathbf{b}_1$
  end for
until $\|\mathbf{b}_1\| \leq \|\mathbf{b}_i\|$ for all $i = 2, \ldots, d$

By induction on d Hermite was able to prove that this algorithm always terminates and the output basis satisfies

$$\prod_{i=1}^{d} \|\mathbf{b}_i\| \leq \sqrt{\gamma_2{}^{d-1}}^{d} \operatorname{vol}(L).$$

So there exists a constant γ_d such that every lattice L of dimension d has some basis $\mathbf{b}_1, \ldots, \mathbf{b}_d$ with $\prod_{i=1}^{d} \|\mathbf{b}_i\| \leq \sqrt{\gamma_d}^{d} \operatorname{vol}(L)$, and Hermite's Algorithm shows $\gamma_d \leq \gamma_2^{d-1}$. In fact, $\gamma_d = \Theta(d)$: for large d, $\frac{d}{2\pi e} < \gamma_d < \frac{d}{\pi e}$.

5 LLL Algorithm

Introduced in 1982 in the context of factoring polynomials, though has since found wider use. It is actually a relaxed version of Hermite's Algorithm: the

condition on line 12 that $\|\mathbf{b}_1\| \leq \|\mathbf{b}_i\|$ for all $i = 2, \ldots, d$ is replaced by the *Lovász condition*

$$\left\|\mathbf{b}_k^* + \mu_{k,k-1}\mathbf{b}_{k-1}^*\right\|^2 \geq \tfrac{3}{4}\left\|\mathbf{b}_{k-1}^*\right\|^2.$$

If $S = \operatorname{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{k-2})^\perp$ then the Lovász condition is just

$$\|\operatorname{proj}_S \mathbf{b}_k\|^2 \geq \tfrac{3}{4}\|\operatorname{proj}_S \mathbf{b}_{k-1}\|^2,$$

so this roughly checks that $\|\mathbf{b}_k\| \geq \|\mathbf{b}_{k-1}\|$, except only with respect to the vector's S-components and with the inequality 'relaxation' coefficient δ which was chosen to be $\frac{3}{4}$ in the original LLL paper. Increasing δ may improve the returned basis at the expense of runtime, though LLL is known to run in polynomial time for all $\delta \in (\frac{1}{4}, 1)$. Specifically, if $\|\mathbf{b}_i\| \leq B$ for all i then LLL takes $O(d^3 n \log B)$ arithmetic operations on integers of size $O(d \log B)$.

Algorithm 3 LLLReduce$(\mathbf{b}_1, \ldots, \mathbf{b}_d)$

Input: A basis $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{R}^n$ for a lattice L.
Output: An LLL-reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_d \in \mathbb{R}^n$ of L.

1. $k = 2$, compute GSO ($\mathbf{b}_i^*$ and $\mu_{i,j}$)
2. **while** $k \leq n$ **do**
3. **for** $i = 1$ to $k - 1$ **do**
4. $\mathbf{b}_i := \mathbf{b}_i - \lfloor \mu_{k,i} \rceil \mathbf{b}_i$, update GSO
5. **end for**
6. **if** $\left\|\mathbf{b}_k^* + \mu_{k,k-1}\mathbf{b}_{k-1}^*\right\|^2 \geq \frac{3}{4}\left\|\mathbf{b}_{k-1}^*\right\|^2$ or $k = 1$ **then**
7. $k := k + 1$
8. **else**
9. swap $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$, update GSO
10. $k := k - 1$
11. **end if**
12. **end while**

This presentation of LLL avoids recursion and computes the GSO components $\mathbf{b}_i^*$ and $\mu_{i,j}$ once at the start of the algorithm and updates them whenever needed.

6 Problems

There are still many open questions concerning lattice reduction:

- Unknown if Hermite's reduction algorithms run in polynomial time, or if LLL runs in polynomial time with $\delta = 1$.
- Is finding the shortest nonzero vector of a lattice NP-hard? Can it be approximated up to a polynomial factor?
- The exact value of Hermite's constant γ_n only known for $n \leq 8$ and $n = 24$.

References

[1] H. Cohen, Number Theory. Vol. I: Tools and Diophantine equations. Graduate Texts in Mathematics, Vol. 239. Springer, Heidelberg (2007)

[2] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:513–534, 1982.

[3] P. Nguyen and D. Stehlé. Floating-point LLL revisited. In *Proceedings of Eurocrypt 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 215–233. Springer-Verlag, 2005.

[4] P. Nguyen and D. Stehlé. An LLL algorithm with quadratic complexity. *In preparation*, 2007.

[5] H. Yao and G. W. Womell. Lattice-Reduction-Aided Detectors for MIMO Communication Systems, in *Proceedings of IEEE Globecom 2002*, Taipei, Taiwan, November 2002.

Lagrange's Algorithm finding a minimal basis