JOURNAL OF ALGEBRAIC STATISTICS
Volume 9, No. 1, 2018, p. 102-108
https://publishoa.com
ISSN: 1309-3452
DOI: https://doi.org/10.52783/jas.v9i1.1445

Guassian Lattice Reduction Algorithm in Two-Dimensions

Shshank Chaube
Department of Mathematics, Graphic Era Hill University, Dehradun, Uttarakhand India 248002

Abstract

An approximation approach for the shortest vector problem, the Lenstra-Lenstra-Lovasz (LLL) Algorithm completes its computations in polynomial time and produces an estimate that is within an exponential factor of the true solution. It's a technique that can be used in practice and is accurate enough to crack cryptosystems, factor polynomials over integers, and solve integer linear programming problems. For the purpose of using the Gauss method "inside" We present a comprehensive examination of the LLL method in the context of a collection of realistic probabilistic models. The proofs focus on both the underlying dynamical systems and the transfer operators. Finding a lattice of Euclidean space with a short basis from a skewed starting point is known as the lattice reduction problem. Especially in cryptology, lattice reduction techniques offer remarkable use in mathematics and computer science.

Keywords: Lenstra-Lenstra-Lovasz, Guassian, Lattice, Reduction Algorithm, Two-Dimension

Introduction

When given an integer lattice basis, reducing a lattice basis is a mathematical technique used to create a foundation with nearly orthogonal short vectors. Several algorithms, each taking at least an execution time exponential in the lattice's dimension, are used to accomplish this. The goal of lattice basis reduction is to transform a given lattice basis into a "good" lattice basis characterized by relatively short, nearly orthogonal vectors. A formal mathematical definition of "lovely basis" and a workable technique for calculating it are required for this. Simply put, a lattice is an infinite set of points in R m that are spaced apart in such a way that any point may be superimposed onto any other point in the set thanks to the symmetry of the arrangement. History difficulties with sphere packings and higher-dimensional adaptations of Euclid's gcd method led to the development of geometry of numbers, the field of number theory concerned with lattices. Iteratively reducing the bigger of the two vectors by adding or removing an integer multiple of the smaller vector is done in the same fashion as the Euclidean algorithm.

The development of a spigot solution for is only one of several recent uses of lattice reduction algorithms in number theory. It is common practice to use LLL while breaking public key cryptosystems. An augmented n×n identity matrix (where the entries in the final column are the n elements multiplied by a large positive constant w to punish vectors that do not sum to zero) is a common kind of input for algorithms that seek integer relations. Integer programming in any given dimension was shown to be possible in polynomial time using the LLL technique for finding a nearly-orthogonal basis.

Literature Review

Stehlé, Damien (2017) Many computing problems may be recast as searching for short non-zero vectors in certain lattices, which is why lattice reduction is so useful. An introduction to lattice reduction algorithms is provided here. The LLL method, which finds relatively short bases in polynomial time, and the BKZ algorithm, which finds shorter bases but at a higher cost, will both be considered. We will use examples from the fplll library to demonstrate the algorithms.

Jazaeri, Shahram & Amiri-Simkooei, A. & Sharifi, Mohammad A. (2014) the best integer solution to the weighted integer least squares problem as quickly as possible is the focus of decorrelation theory or reduction (2014). For a long time, the orthogonality defect was the go-to for gauging how orthogonal the reduced lattice bases were. In this work, we provide a maximum estimate for the possible outcomes of an integer search. The LLL, LAMBDA, MLAMBDA, and Seysen decorrelation algorithms—four of the most popular in the field—are studied and contrasted. The Seysen reduction method is superior to the alternatives in reducing the condition number, as shown by extensive testing on both generated data with various condition numbers and dimensions and actual GPS data. All methods are measured for both their initial and final totals of integer candidates. Integer candidate counts, condition numbers, and orthogonality defects are all compared to one another, and it is shown that lowering either the condition numbers or the defect may not always result in fewer integer candidates. Contrary to popular belief, the results show that LAMBDA and MLAMBDA perform significantly better in reducing the number of integer candidates for both the general integer least squares estimation issue and the integer ambiguity resolution problem, even if they have a larger orthogonality defect and condition number in certain circumstances.

Dias, Sérgio & Vieira, Newton (2017) According to this explanation, formal concept analysis (FCA) is a mathematical theory of data analysis with several useful applications. Several fields have identified the challenge of building a large enough idea lattice as a significant barrier to progress in FCA. Many methods for idea lattice reduction were developed, each with its own set of features for dealing with this issue. The question of what kinds of knowledge changes may come from a reduction, however, cannot be adequately addressed by existing methodologies. Here, we describe an approach to analysis that makes use of idea lattice reduction. It relies on a set of correct implications that are true in both the full and reduced formal settings, often known as idea lattices. The approach uses both sets of implications to demonstrate what aspects of a reduction are maintained, lost, gained, or altered. We examine three categories of reduction methods from a methodological perspective, drawing out the shared characteristics between them based on the nature of the changes they do. In each lesson, students will apply their newfound knowledge to a concrete case.

Lattice Reduction in Dimension 2

Here we provide both the (initial) vectorial and the complex versions of the Gauss methods. We describe the most important parameters and how they affect the LLL algorithm analysis.

By making a tiny linguistic stretch, we may write a complex number $z\in \mathbb{C}$ in the same way as the vector of R^2 whose components are $(\Re z, \Im z)$. The modulus and Euclidean norm of a complex vector are both denoted by the symbol $|z|$, while the scalar product of two complex vectors is written as (u .v).

$$\frac{v}{u} = \frac{(u \cdot v)}{|u|^2} + i\frac{\det(u,v)}{|u|^2} \quad (1)$$

Some of the lattice L's bases, termed reduced bases, have the advantage of being "short" vector bases. Minimal bases with optimality features are the best reduced bases in dimension 2: For a given lattice L, we say that the non-zero vector u with the shortest Euclidean norm is a first minimum of L, and we write the length of a first minimum of L as $\lambda_1(L)$. Every vector among the lattice's shortest vectors that is linearly independent on u is a second minimum v, and its Euclidean length is indicated as $\lambda_2(L)$. If a basis includes both the first and second minimums, we say that it is minimal. Refer to Figure 1. In what follows, we zero down on certain bases that have one of two characteristics:

(P) the determinant is greater than zero ($\det(u, v) \geq 0$). To have a good foundation is to have a solid foundation.

(A) the scalar product is non-negative (i.e., $(u. v) \geq 0$). An acute foundation is a solid one.

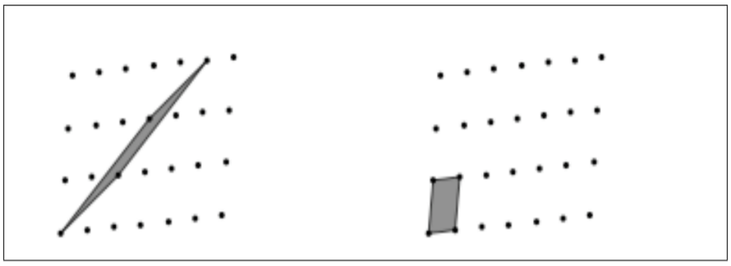


Figure 1: A lattice and two of its bases represented by the parallelogram they span.

Because one of (u, v) and (u, -v) is, we may always, without introducing a special case, assume that a basis is acute (resp. positive). The next finding characterizes minimalistic bases. The evidence is missing.

Proposition 1. [Descriptions of rudimentary foundations.] (P) Positive foundations Consider the positive basis (u, v). If (a) and (b) below are equal, then (a) and (b) are true.

- (u, v) is a minimal basis;
- (u, v) simultaneously fulfills the three inequalities:

$$(P_1): \quad |\frac{v}{u}| \geq 1, \quad (P_2): \quad |\Re(\frac{v}{u})| \leq \frac{1}{2} \qquad and \quad (P_3): \quad \Im(\frac{v}{u}) \geq 0$$

(A) [Sharp acid bases.] An acute basis would be (u, v). If (a) and (b) below are equal, then (a) and (b) are true.

$$(A_1): \quad |\frac{v}{u}| \geq 1, \qquad and \qquad (A_2): \quad 0 \leq \Re(\frac{v}{u}) \leq \frac{1}{2}.$$

The Gaussian reduction schemes

According on whether one prefers to work with positive bases or acute bases, there are two distinct reduction techniques.

The positive Gauss Algorithm

When given a positive arbitrary basis, the positive lattice reduction process returns a positive minimum basis. The goal of the positive Gauss algorithm is to fulfill all of the requirements (P) of Proposition 1 at the same time. Exchange of vectors followed by sign change v:= −v satisfies both (P1) and (P3). Integer representations of the types satisfy the requirement (P2):

$$v := v - mu \qquad \text{with} \quad m := \lfloor r(v,u) \rceil, \quad r(v,u) := \Re(\frac{v}{u}) = \frac{(u \cdot v)}{|u|^2}$$

The last pair (vp, vp+1) meets the requirements (P) of Proposition Z, p ≡ p(u, v), where p is an integer in the range [0, 1].

```
PGauss$(u, v)$
Input. A positive basis $(u, v)$ of $\mathbb{C}$ with $|v| \leq |u|$, $|r(v,u)| \leq (1/2)$.
Output. A positive minimal basis $(u, v)$ of $\mathcal{L}(u, v)$ with $|v| \geq |u|$.
While $|v| \leq |u|$ do
        $(u, v) := (v, -u)$;
        $m := \lfloor r(v,u) \rceil$, with $r(v,u) = \frac{(u \cdot v)}{|u|^2}$;
        $v := v - mu$;
```

1. A unimodular matrix Mi is defined at each stage with detMi = 1.

$$\mathcal{M}_i = \begin{pmatrix} m_i & -1 \\ 1 & 0 \end{pmatrix}, \qquad \text{with} \quad \begin{pmatrix} v_{i+1} \\ v_i \end{pmatrix} = \mathcal{M}_i \begin{pmatrix} v_i \\ v_{i-1} \end{pmatrix}$$

Suppose a matrix M is generated using the algorithm for which

$$\begin{pmatrix} v_{p+1} \\ v_p \end{pmatrix} = \mathcal{M} \begin{pmatrix} v_1 \\ v_0 \end{pmatrix} \qquad \text{with} \quad \mathcal{M} := \mathcal{M}_p \cdot \mathcal{M}_{p-1} \cdot \ldots \cdot \mathcal{M}_1. \tag{3}$$

Lattice Reduction Algorithms

Many cryptosystems have been shown, the safety of which is contingent on the difficulty of solving apprSVP and/or apprCVP in different lattices. A solution to these difficulties is described below using an algorithm called LLL, which can achieve an accuracy of within a factor of C^n, where C is a tiny constant and n is the dimension of the lattice. Hence, the LLL algorithm almost solves SVP and CVP in low dimensions but fails miserably in high dimensions.

Gaussian Lattice Reduction in Dimension 2

Gauss is largely responsible for the procedure used to determine the best possible lattice basis in dimension 2. The idea is simple: wherever possible, try to gain ground by subtracting multiples of one basis vector from the other. Let us now suppose that there are two basis vectors, v1 and v2, in the lattice L R2. If we swap v1 and v2 around, we can say that ||v1|| > ||v2||. To reduce v2, we remove a number that is a power of v1. The vector v2 may be used in lieu of v1 if we were permitted to remove any integer multiple of v1.

$$\boldsymbol{v}_2^* = \boldsymbol{v}_2 - \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\|\boldsymbol{v}_1\|^2}\boldsymbol{v}_1$$

which is perpendicular to v1.

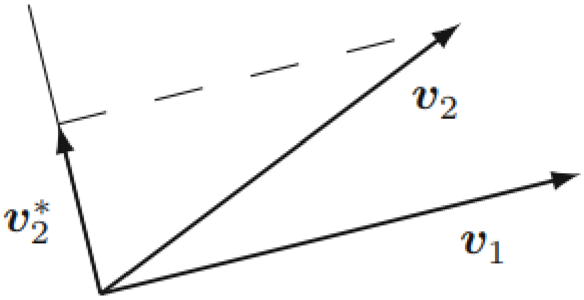


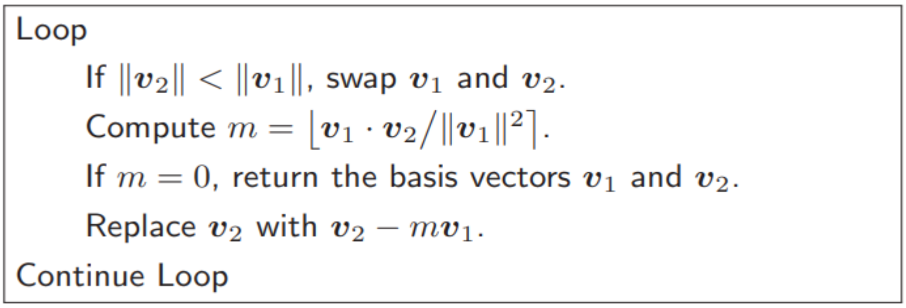
Figure 2: The expression v 2 represents the orthonormal projection of vector v2 onto vector v1.

Since the vector v2 is highly unlikely to be in L, this is obviously dishonest. Subtracting v1 from v2 is restricted to integer multiples only. Hence, we substitute the vector v2 for v2 as best we can.

$$\boldsymbol{v}_2 - m\boldsymbol{v}_1 \qquad \text{with} \quad m = \left\lfloor \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\|\boldsymbol{v}_1\|^2} \right\rceil$$

We will halt if v2 is still longer than v1. If this is not the case, we will switch v1 and v2. Gauss demonstrated that the process ends, and that the resultant L basis is excellent. The following proposal clarifies this.

Proposition 2 (Gaussian Lattice Reduction). Consider the 2-dimensional lattice $L \subset R^2$ to have v1 and v2 as its basis vectors. The following algorithm is guaranteed to finish and provide a solid L-base.



```
Loop
    If $\|v_2\| < \|v_1\|$, swap $v_1$ and $v_2$.
    Compute $m = \lfloor v_1 \cdot v_2 / \|v_1\|^2 \rceil$.
    If $m = 0$, return the basis vectors $v_1$ and $v_2$.
    Replace $v_2$ with $v_2 - mv_1$.
Continue Loop
```

In particular, SVP is solved if and only if the terminating vector v1 is the shortest non-zero vector in L. Thus, θ in v1 and v2 have an angle that satisfies | cos θ|≤||v1||/2||v2||, so in specific, π/3 ≤ θ ≤ 2π/3.

Proof. We just show that v1 is the smallest non-zero lattice vector and let the reader fill in the rest. Now, let's assume the algorithm has completed and given us back two vectors, v1 and v2. Therefore, ||v2|| ≥||v1||, and

$$\frac{|\boldsymbol{v}_1 \cdot \boldsymbol{v}_2|}{\|\boldsymbol{v}_1\|^2} \leq \frac{1}{2}. \quad (4)$$

Condition (4) states (geometrically) that reducing v2 by removing an integral multiple of v1 from v2 is impossible. Let's pretend now that any non-zero vector in L is v ∈L. Writing

v = a1v1 + a2v2 with a1, a2 ∈ Z,

we discover that

$$\begin{aligned}\|\boldsymbol{v}\|^2 &= \|a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2\|^2 \\ &= a_1^2\|\boldsymbol{v}_1\|^2 + 2a_1a_2(\boldsymbol{v}_1 \cdot \boldsymbol{v}_2) + a_2^2\|\boldsymbol{v}_2\|^2 \\ &\geq a_1^2\|\boldsymbol{v}_1\|^2 - 2|a_1a_2|\,|\boldsymbol{v}_1 \cdot \boldsymbol{v}_2| + a_2^2\|\boldsymbol{v}_2\|^2\end{aligned}$$

$$\geq a_1^2\|\boldsymbol{v}_1\|^2 - |a_1a_2|\|\boldsymbol{v}_1\|^2 + a_2^2\|\boldsymbol{v}_2\|^2 \quad \text{from (4),}$$

$$\begin{aligned}&\geq a_1^2\|\boldsymbol{v}_1\|^2 - |a_1a_2|\|\boldsymbol{v}_1\|^2 + a_2^2\|\boldsymbol{v}_1\|^2 \quad \text{since } \|\boldsymbol{v}_2\| \geq \|\boldsymbol{v}_1\| \\ &= \left(a_1^2 - |a_1|\,|a_2| + a_2^2\right)\|\boldsymbol{v}_1\|^2.\end{aligned}$$

The amount for any two real values t1 and t2 is

$$t_1^2 - t_2t_2 + t_2^2 = \left(t_1 - \frac{1}{2}t_2\right)^2 + \frac{3}{4}t_2^2 = \frac{3}{4}t_1^2 + \left(\frac{1}{2}t_1 - t_2\right)^2$$

unless both t1 = t2 = zero. Since both a1 and a2 are non-zero integers, we may conclude that $||v||^2$ $\geq||v1||^2$. That v1 is the smallest nonzero vector in L is shown here.

LLL Basis Reduction

Ajtai demonstrated in 1997 that SVP is NP-hard to solve precisely under randomized reduction, while Micciancio demonstrated in 2002 that SVP is NP-hard to approximate with any factor less than √2. A fair approximation of the SVP might be valuable in practical issues, despite the fact that it is shown intractable within a realistic time frame. Based on what we saw in the two-dimensional situation, we learned that making the basis vectors as orthogonal as feasible is preferable throughout the process of basis reduction. There is no more reduction possible in the orthogonal basis. In 1982, A. Lenstra, H. Lenstra, and L. Lov'asz presented the LLL basis reduction method as an approximation to the Gram-Schmidt orthogonalization in higher dimensions. First, they established the meaning of the LLL reduced basis.

Definition 2 (LLL reduced basis). It can be shown that ui,k $=\frac{bk\cdot bi *}{* bi * \cdot bi *}$holds for any n-dimensional Lattice L and any orthogonal basis produced from it using **Theorem 1.** If the following two requirements are met, we declare that the Let {b1, b2, · · ·, bn} is an LLL reduced basis.

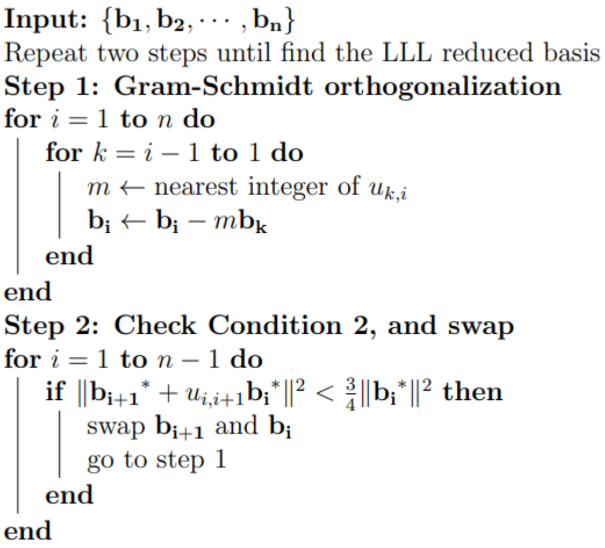
(1) ∀i≠k, ui,k $\leq \frac{1}{2}$

(2) For each i, ||bi+1 $*$ + ui,i+1bi$*$|| $^{2} \geq \frac{3}{4}$ ||bi $*$ ||^{2}

Remark. For the sake of brevity, we'll use the constant $\frac{3}{4}$. The algorithm's termination may be guaranteed in polynomial time with any constant between $\frac{1}{4}$ and 1.

The LLL method for obtaining an LLL reduced basis from a basis {b1, b2, · · · ,bn} in n-dimensions works as follows.

Algorithm 1: LLL Algorithm



```
Input: $\{\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n}\}$
Repeat two steps until find the LLL reduced basis
Step 1: Gram-Schmidt orthogonalization
for $i = 1$ to $n$ do
    for $k = i - 1$ to $1$ do
        $m \leftarrow$ nearest integer of $u_{k,i}$
        $\mathbf{b_i} \leftarrow \mathbf{b_i} - m\mathbf{b_k}$
    end
end
Step 2: Check Condition 2, and swap
for $i = 1$ to $n - 1$ do
    if $\|\mathbf{b_{i+1}}^* + u_{i,i+1}\mathbf{b_i}^*\|^2 < \frac{3}{4}\|\mathbf{b_i}^*\|^2$ then
        swap $\mathbf{b_{i+1}}$ and $\mathbf{b_i}$
        go to step 1
    end
end
```

Step one included finding the most orthogonal basis using Gram-Schmidt orthogonalization, and step two involved verifying our second condition. We switch the bases and go back to step 1 if any of them are out of order. In polynomial time, within an exponential factor, the LLL method gives an estimate of the shortest vector. Initially, we show that the shortest vector is very close to the reduced basis produced by the LLL Algorithm.

Conclusion

We introduced the LLL basis reduction algorithm, a method for approximating the shortest vector in a higher dimensional space that may be executed in polynomial time. The algorithm's efficiency and precision have ensured its continued usage in fields such as number theory, integer programming, and cryptography since its creation in the 1980s. As the previous encryption techniques may be readily cracked by the LLL algorithm, its debut sparked fresh research of cryptography. For lattice problems specifically, it is also a basic algorithm. Gauss is largely responsible for the procedure used to determine the best possible lattice basis in dimension 2. The basic concept is to repeatedly subtract multiples of one basis vector from the other until no more gains can be made.

References

1. Stehlé, Damien. (2017). Lattice Reduction Algorithms. 11-12. 10.1145/3087604.3087665.
2. Neumaier, Arnold & Stehlé, Damien. (2016). Faster LLL-type Reduction of Lattice Bases. 373-380. 10.1145/2930889.2930917.
3. Jazaeri, Shahram & Amiri-Simkooei, A. & Sharifi, Mohammad A. (2014). On lattice reduction algorithms for solving weighted integer leastsquares problems: Comparative study. GPS Solutions. 18. 10.1007/s10291-013-0314-z.
4. Dias, Sérgio & Vieira, Newton. (2017). A methodology for analysis of concept lattice reduction. Information Sciences. 396. 10.1016/j.ins.2017.02.037.
5. Yunus, Umut & Hamdulla, Askar & Jia, Zhen & Ubul, Kurban. (2015). Lattice Reduction Algorithm Based Uplink MC-CDMA. Applied Mechanics and Materials. 738-739. 391-396. 10.4028/www.scientific.net/AMM.738-739.391.
6. Plouffe, Simon. (2013). The lattice reduction algorithm and applications (LLL and PSLQ).
7. Ranjitham, G. & Kumar, K.R.S. (2014). "A novel low complexity diagonal reduction algorithm of lattice reduction for signal detection in Mimo receiver". Journal of Theoretical and Applied Information Technology. 60. 581-586.
8. Domene, Fernando & Jozsa, Csaba & Vidal, Antonio & Pinero, Gema & Gonzalez, Alberto. (2013). Performance analysis of a parallel lattice reduction algorithm on many-core architectures.
9. Liao, Chun-Fu & Chai, Li-Wei & Huang, Yuan-Hao. (2012). Loop-Reduction LLL Algorithm and Architecture for Lattice-Reduction-Aided MIMO Detection. Journal of Electrical and Computer Engineering. 2012. 10.1155/2012/876380.
10. Saruchi, & Morel, Ivan & Stehlé, Damien & Villard, Gilles. (2014). LLL reducing with the most significant bits. Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC. 10.1145/2608628.2608645.