Some Connections Between Learning Integer Lattices with Noise and Learning Parity Functions with Noise

Alexander D. Healy
ahealy@fas.harvard.edu

Abstract

We consider the problem of learning integer lattices in the presence of either classification noise or malicious noise, and show that learning integer lattices is at least as hard as learning $\mathbb{F}_q$-linear subspaces over any finite field, $\mathbb{F}_q$. In particular, learning integer lattices with noise is as hard as learning a parity function with noise. We also give a converse result which shows that the ability to learn linear subspaces of co-dimension 1 over $\mathbb{F}_p$ is sufficient to learn a restricted class of lattices, provided that the factorization of the lattice's determinant is given.

Introduction

A major open problem in learning theory is that of PAC-learning a hidden parity function in the presence of classification noise. The problem of learning integer lattices is very closely related to that of learning parity functions, or more generally, learning linear subspaces of $\mathbb{F}_q^n$. Indeed, the most natural mistake-bounded algorithms for learning each of these are very similar, and more generally there is a long history of intimate connections between the geometry of lattices and the geometry of Hamming space [CS93]. For these reasons, it is natural to consider the complexity of learning integer lattices in the presence of noise, as this might shed some light on the complexity of learning parity in the presence of noise.

In the sequel, we recall (in section 1) basic definitions and properties of lattices, we discuss (in section 2) known algorithms for learning linear subspaces and lattices without noise, we show (in section 3) that learning integer lattices in the presence of classification (respectively, malicious) noise, is as hard as learning $\mathbb{F}_q$-linear subspaces of $\mathbb{F}_q^n$ in the presence of classification (resp. malicious) noise, and in particular, as hard as learning parity in the presence of classification (resp. malicious) noise, and finally we give a converse result (in section 4) which shows that the ability to learn linear subspaces of co-dimension 1 over $\mathbb{F}_p$ is sufficient to learn a restricted class of lattices, provided that the factorization of the lattice's determinant is given. We conclude with a discussion of the consequences of these results, and possible directions for future work.

1 Lattices

A *lattice* is a discrete subgroup $\mathcal{L}$ of n-dimensional Euclidean space $\mathbb{R}^n$ given by $\mathcal{L} = \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_m$, where the vectors $b_1, \ldots, b_m \in \mathbb{R}^n$ are linearly independent. An *integer lattice* is a lattice where $b_i \in \mathbb{Z}^n$ for all i (or, equivalently $\mathcal{L} \subseteq \mathbb{Z}^n$), and in this paper we are primarily concerned with integer lattices, so we shall use the terms interchangeably. The $b_1, \ldots, b_m$ are said to form a *basis* for the lattice $\mathcal{L}$, and typically a lattice is given by an $n \times m$ matrix of the form:

$$B = \begin{pmatrix} \begin{bmatrix} | \\ b_1 \\ | \end{bmatrix} & \begin{bmatrix} | \\ b_2 \\ | \end{bmatrix} & \cdots & \begin{bmatrix} | \\ b_m \\ | \end{bmatrix} \end{pmatrix}$$

This allows us to think of $\mathcal{L}$ as the image of $\mathbb{Z}^m$ under the linear map defined by B, i.e. $\mathcal{L} = B\mathbb{Z}^m$. However, the basis of a lattice is not unique; indeed, if M is an $m \times m$ integer matrix with determinant ± 1, a so-called *unimodular* matrix, then it is not hard to verify that BM is also a basis for $\mathcal{L}$ and, conversely, that every basis for $\mathcal{L}$ is of the form BM for some unimodular M. It is well-known that the *determinant* of the lattice $\mathcal{L}$, given by $\det(\mathcal{L}) = \sqrt{\det(B^T B)}$, measures the m-dimensional volume of the *fundamental parallelepiped* of $\mathcal{L}$,

$$P(B) = \{c_1\vec{b}_1 + \cdots + c_m\vec{b}_m \in \mathbb{R}^n \mid 0 \leq c_i < 1\}$$

and is an invariant of the lattice; that is, the determinant does not depend on the choice of basis B.

Another important fact is that $\mathcal{L}$ is a lattice, and $\mathcal{L}' \subseteq \mathcal{L}$ is a *sublattice* of $\mathcal{L}$, then the index of $\mathcal{L}'$ in $\mathcal{L}$ (as a subgroup), is given by $[\mathcal{L} : \mathcal{L}'] = |\mathcal{L} \cap P(B')| = \det(\mathcal{L}')/\det(\mathcal{L})$.

Finally, a lattice is said to be *full-dimensional* if its basis consists of n linearly independent vectors, i.e. if $\mathcal{L}$ is not contained in any proper subspace of $\mathbb{R}^n$.

For more details about lattices and relevant algorithms, we refer the reader to [CS93, Coh93].

2 Learning Parity and Lattices without Noise

We say that $f : \{0,1\}^n \to \{0,1\}$, is a parity function if

$$f(x_1, \ldots, x_n) = \bigoplus_{i \in S} x_i$$

for some *non-empty* subset $S \subseteq \{1, 2, \ldots, n\}$, and we call vectors $x = (x_1, \ldots, x_n)$, such that $f(x) = 0$, *positive* examples. Thus, the positive examples for a parity function form a linear subspace of $\mathbb{F}_2^n$ of co-dimension 1 (i.e., dimension $n - 1$); indeed, it is not hard to see that the positive examples are closed

under addition modulo 2, and since precisely half of the vectors $x \in \{0,1\}^n$ are positive examples, the dimension of this space is necessarily $n-1$.

The standard algorithm for learning parity is the following mistake-bounded algorithm. In fact, the algorithm naturally generalizes to learning arbitrary linear spaces $S \subseteq \mathbb{F}^n$ over any finite field $\mathbb{F}$, so we shall give this more general algorithm.

Algorithm 2.1.
Begin with an empty basis $B = \{\}$
For each example $x \in \mathbb{F}^n$,
    If $x \in \text{span}(B)$ *then predict* $x \in S$, *else predict* $x \notin S$
    If $x \in S$, *but* $x \notin \text{span}(B)$, *set* $B = B \cup \{x\}$

The above algorithm clearly never makes any false-positive predictions since the prediction $x \in S$ is only ever made if x can be expressed as a linear combination of vectors that were guaranteed to be in S by the example oracle, i.e. only if $x \in S$. Every time a false-negative prediction is made, the example x is added to the basis B, increasing the rank of B by one (since, by assumption, x is not in the span of B). Therefore at most n false-negative predictions can be made.

This learning algorithm is also efficient, as it only ever requires the solution to a linear system over $\mathbb{F}$, which can be computed in polynomial-time [Coh93].

It is well-known that mistake-bounded algorithms with polynomial mistake bounds, such as this one, yield polynomial-time, PAC-learning algorithms. Therefore, linear subspaces of $\mathbb{F}_p^n$ are PAC-learnable.

We will now turn our attention to the algorithm for learning integer lattices, but first we need to define the learning problem at hand.

Definition 2.2. *The problem of PAC-learning a hidden integer lattice,* $\mathcal{L}$, *is the following. Given* ϵ, δ, m, *and an example oracle providing examples* $x \in \mathbb{Z}^n$ *and their classification (whether* $x \in \mathcal{L}$*) drawn from a distribution* D *over* $[-m, m]^n$, *learn, in time* $\text{poly}(1/\epsilon, 1/\delta, \log m, n)$, *a hypothesis* h *that with probability* $1-\delta$ *has error at most* ϵ *on examples drawn from the distribution* D.

A very natural mistake-bounded algorithm for learning integer lattices is the following, which is essentially the algorithm given in [HSW92].

Algorithm 2.3.
Begin with an empty basis $B = \{\}$
For each example $x \in \mathbb{Z}^n$,
    If x *is contained in,* $\langle B \rangle$, *the lattice generated by* B, *then predict* $x \in \mathcal{L}$, *else predict* $x \notin \mathcal{L}$
    If $x \in \mathcal{L}$, *but* $x \notin \langle B \rangle$, *then*
        set $B = B \cup \{x\}$ *and replace* B *by an equivalent basis of linearly independent vectors*

As with the previous algorithm, this algorithm will clearly never make false-positive predictions. When a false-negative prediction is made, there are two cases: Either (1) $x \notin \text{span}(B)$, in which case the rank of the basis increases by one, so there can be at most n such false-negatives, or (2) $x \in \text{span}(B)$ but $x \notin \langle B \rangle$, in which case $\mathcal{L}' = \langle B \rangle$ is a proper sublattice of the lattice $\mathcal{L}'' = \langle B \cup \{x\} \rangle$, so it must be the case that $\det(\mathcal{L}'') \leq \det(\mathcal{L}')/2$ (since $\det(\mathcal{L}')/\det(\mathcal{L}'') = [\mathcal{L}'' : \mathcal{L}'] > 1$). Each time a false-negative of the first type is made, the determinant can increase by at most a factor of $m\sqrt{n}$ (the length of the longest vector in $[-m, m]^n$), and the first false-negative results in the determinant changing from 0 to some value that is at most $m\sqrt{n}$, thus the determinant of $\langle B \rangle$ can never increase by more than than a total of $(m\sqrt{n})^n$. On the other hand, the determinant of $\langle B \rangle$ must always be the square-root of a positive integer (except when the algorithm begins), so the number of false-negatives of the second type (which each reduce the determinant by at least a factor of 2) is at most $\log_2((m\sqrt{n})^n) = n \log_2(m\sqrt{n})$. This gives a mistake bound of $n + n \log_2(m\sqrt{n})$.

Also, this algorithm is efficient: testing if $x \in \langle B \rangle$ simply involves solving the real linear system $By = x$, and checking if y is an integer vector. Additionally, computing a basis B' that is equivalent to $B \cup \{x\}$, but that consists of linearly independent vectors, can be performed in polynomial time, using an efficient algorithm for computing the so-called Hermite Normal Form of a lattice, [Coh93, MW00, HSW92].

As before, such a polynomial mistake bound implies the existence of a PAC-learning algorithm for learning a hidden integer lattice.

Another definition whose importance will become apparent later, is the following restricted notion of learning a hidden lattice.

Definition 2.4. *The problem of PAC-learning a hidden full-dimensional, square-free integer lattice, $\mathcal{L}$, is the following. Provided that $\mathcal{L}$ is a full-dimensional lattice, that* $\det(\mathcal{L})$ *is square-free (i.e., not divisible by any square integer), and that the factorization of* $\det(\mathcal{L})$ *is given to the learning algorithm, then given ϵ, δ, m, and an example oracle providing examples $x \in \mathbb{Z}^n$ and their classification (whether $x \in \mathcal{L}$) drawn from a distribution D over $[-m, m]^n$, learn, in polynomial time, a hypothesis h that with probability $1 - \delta$ has error at most ϵ on examples drawn from the distribution D.*

3 Learning $\mathbb{F}_q$-Linear Subspaces via Lattices

Although it will be subsumed by a more general result, we first give a simple reduction that shows that learning an integer lattice under any noise conditions is as hard as learning a hidden parity function under the same noise conditions.

Proposition 3.1. *There is a PAC-learning reduction from learning parity to learning integer lattices, using the same examples.*

Proof. Let P denote the collection of positive examples of the hidden parity function. Now let y be an arbitrary vector in $\{0,1\}^n$ and define a lattice

$$\mathcal{L} = \{v \in \mathbb{Z}^n \mid v \equiv x \bmod 2, \text{ for some } x \in P\}$$

where $v \equiv x \bmod 2$ means that $v_i \equiv x_i \bmod 2$ in each coordinate. Clearly, if $y \in P$, then $y \in \mathcal{L}$. Conversely, if $y \in L$, then $y \equiv x \pmod 2$ for some $x \in P$, and since $P \subseteq \{0,1\}^n$ and $y \in \{0,1\}^n$, we have $y = x$, and hence $y \in P$. Therefore, $\mathcal{L} \cap \{0,1\}^n = P$, so the ability to learn $\mathcal{L}$ under any distribution restricted to $\{0,1\}^n$ implies the ability to learn parity under any distribution. □

In fact, the above proof generalizes to any $\mathbb{F}_p$-linear subspace, $S \subseteq \mathbb{F}_p^n$, by using the lattice

$$\mathcal{L} = \{v \in \mathbb{Z}^n \mid v \equiv x \bmod p, \text{ for some } x \in S\}$$

yielding the following more general reduction.

Proposition 3.2. *There is a PAC-learning reduction from learning $\mathbb{F}_p$-linear subspaces of $\mathbb{F}_p^n$ to learning integer lattices, using the same examples.*

At this point, we note that if we restrict our attention to the problem of learning an $\mathbb{F}_p$-linear subspace, $S \subseteq \mathbb{F}_p^n$, of co-dimension 1, by analogy with parity which is a linear subspace of $\mathbb{F}_2^n$ of co-dimension 1, then the resulting lattice

$$\mathcal{L} = \{v \in \mathbb{Z}^n \mid v \equiv x \bmod p, \text{ for some } x \in S\}$$

has determinant $\det(\mathcal{L}) = p$. Indeed, the subspace S contains p^{n-1} points, so by the above argument, we have $|\mathcal{L} \cap [0,p)^n| = p^{n-1}$. On the other hand, we have that $p\mathbb{Z}^n$ is a sublattice of $\mathcal{L}$, and hence $[\mathcal{L} : p\mathbb{Z}^n] = \det(p\mathbb{Z}^n)/\det(\mathcal{L}) = p^n/\det(\mathcal{L})$. Since $[\mathcal{L} : p\mathbb{Z}^n] = |\mathcal{L} \cap [0,p)^n| = p^{n-1}$, it must be the case that $\det(\mathcal{L}) = p$.

We also note that the lattice $\mathcal{L}$ is *full-dimensional*, as it contains n linearly independent vectors (e.g., $\{(p,0,0,\ldots,0),(0,p,0,\ldots,0),\ldots,(0,0,0,\ldots,p)\}$). Thus, since p is known (implicitly) to the learning algorithm, the problem of PAC-learning $\mathbb{F}_p$-linear subspaces of co-dimension 1 reduces to the problem of PAC-learning a hidden full-dimensional, square-free integer lattice, a possibly easier problem than that of PAC-learning arbitrary integer lattices.

Finally, we conclude this section by noting that learning $\mathbb{F}_q$-linear subspaces of $\mathbb{F}_q^n$, where $q = p^k$ reduces to learning linear subspaces of $\mathbb{F}_p^{kn}$, and consequently reduces to learning integer lattices. Indeed, if we recall that $\mathbb{F}_q \simeq \mathbb{F}_p[x]/(f(x))$, where $f(x) \in \mathbb{F}_p[x]$ is an irreducible polynomial of degree k, then given an $\mathbb{F}_q$-linear subspace $S \subseteq \mathbb{F}_q^n$, there is a basis $B = \{b_1,\ldots,b_d\}$ for S such that $S = \mathbb{F}_q b_1 + \cdots + \mathbb{F}_q b_d$. Therefore, since $\mathbb{F}_q = \mathbb{F}_p + \mathbb{F}_p x + \cdots + \mathbb{F}_p x^{k-1} \subseteq \mathbb{F}_p[x]/(f(x))$, we have that $B \cup Bx \cup \cdots \cup Bx^{k-1}$ is basis for S as an $\mathbb{F}_p$-linear subspace of $\mathbb{F}_q^n$, and $\mathbb{F}_q^n \simeq \mathbb{F}_p^{kn}$ as $\mathbb{F}_p$-linear vector spaces. Therefore, despite their more complex appearance, learning $\mathbb{F}_q$-linear subspaces of $\mathbb{F}_q$ reduces to learning $\mathbb{F}_p$-linear subspaces of $\mathbb{F}_p^{kn}$.

4 Learning Lattices via $\mathbb{F}_p$-Linear Subspaces

In this section we focus on reductions from learning lattices to learning $\mathbb{F}_p$-linear subspaces of $\mathbb{F}_p^n$. We will conclude by showing that learning full-dimensional square-free integer lattices in the presence of classification (respectively malicious) noise is equivalent to learning co-dimension 1 $\mathbb{F}_p$-linear subspaces of $\mathbb{F}_p^n$ in the presence of classification (resp. malicious) noise.

In the previous section, we noted that the problem of learning a co-dimension 1 subspaces of $\mathbb{F}_p^n$ in the presence of noise reduces to the problem of learning a full-dimensional lattice of determinant p in the presence of noise. Thus, we have that learning co-dimension 1 subspaces of $\mathbb{F}_p^n$ in the presence of noise reduces to the problem of learning a full-dimensional square-free lattice in the presence of noise. For the remainder of this section we focus on the converse, i.e. showing that learning full-dimensional square-free lattices reduces to learning co-dimension 1 subspaces of $\mathbb{F}_p^n$. The reduction is given by the following algorithm.

Algorithm 4.1.
Given: $\det(\mathcal{L})$ *and its factorization* $\det(\mathcal{L}) = p_1 p_2 \cdots p_r$.

Initialize a learning algorithm, A_i*, that learns co-dimension* 1 *subspaces* $S_i \subseteq \mathbb{F}_{p_i}^n$ *for each* $1 \leq i \leq r$*, with error* ϵ/r *and confidence* $1 - \delta/r$.

For every example $x \in \mathbb{Z}^n$ *and its classification* $c \in \{0, 1\}$*, pass* $(x \bmod p_i, c)$ *to algorithm* A_i.

To predict on $x \in \mathbb{Z}^n$*, pass* $x \bmod p_i$ *to* A_i*, and respond* $x \in \mathcal{L}$ *if and only if all* A_i*'s predict* $x \in S_i$.

The following lemmas will be necessary to prove the correctness of this algorithm.

Lemma 4.2. *Let* $\mathcal{L}$ *be a full-dimensional integer lattice, and let* $x \in \mathbb{Z}^n$*. Then* $x \in \mathcal{L}$ *if and only if* $x \equiv y \bmod \det(\mathcal{L})$*, for some* $y \in \mathcal{L}$.

Proof. We begin by showing that $\det(\mathcal{L})\mathbb{Z}^n \subseteq \mathcal{L}$. Let B be a basis matrix for $\mathcal{L}$, i.e. such that $\mathcal{L} = B\mathbb{Z}^n$. Since $\mathcal{L}$ is a full-dimensional integer lattice, B is a square integer matrix and $\det(\mathcal{L}) = |\det(B)|$. It is well-known that $|\det(B)|B^{-1}$ is also an integer matrix, for example by "Cramer's rule" from linear algebra; thus, if we take an arbitrary $z \in \det(\mathcal{L})\mathbb{Z}^n$, we have that $B^{-1}z = |\det(B)|B^{-1}z'$ for some $z' \in \mathbb{Z}$, and so it is an integer vector, as $|\det(B)|B^{-1}$ is an integer matrix. Therefore, we may conclude that $z = B(B^{-1}z) \in B\mathbb{Z}^n = \mathcal{L}$, that is $z \in \mathcal{L}$, and since z was an arbitrary vector in $\det(\mathcal{L})\mathbb{Z}^n$, we have $\det(\mathcal{L})\mathbb{Z}^n \subseteq \mathcal{L}$.

Finally, suppose we have a vector $x \equiv y \bmod \det(\mathcal{L})$, for some $y \in \mathcal{L}$. Then $x = y + v$ for some $v \in \det(\mathcal{L})\mathbb{Z}^n \subseteq \mathcal{L}$; since $y \in \mathcal{L}$ and $v \in \mathcal{L}$, it must be the case that $x = y + v \in \mathcal{L}$ as well. □

Lemma 4.3. *Let $\mathcal{L}$ be a full-dimensional integer lattice with square-free determinant,* $\det(\mathcal{L}) = p_1 p_2 \cdots p_r$*, and let $x \in \mathbb{Z}^n$. Then $x \in \mathcal{L}$ if and only if for all primes p_i dividing* $\det(\mathcal{L})$*, we have $x \equiv y_i \bmod p_i$, for some $y_i \in \mathcal{L}$.*

Proof. By applying the Chinese Remainder Theorem to each coordinate separately, we know there is a natural bijection

$$\varphi : \frac{\mathbb{Z}^n}{\det(\mathcal{L})\mathbb{Z}^n} \to \mathbb{F}_{p_1}^n \times \cdots \times \mathbb{F}_{p_r}^n$$

defined by $\varphi(x_1, \ldots, x_n) \mapsto (\varphi_1(x_1, \ldots, x_n), \ldots, \varphi_r(x_1, \ldots, x_n))$, where

$$\varphi_i(x_1, \ldots, x_n) = (x_1 \bmod p_i, \ldots, x_n \bmod p_i) \in \mathbb{F}_{p_i}^n$$

Clearly, the image of $\mathcal{L}/\det(\mathcal{L})\mathbb{Z}^n$ under φ_i is a linear subspace consisting exactly of those elements of $\mathbb{F}_{p_i}$ that are congruent to some lattice vector modulo p_i. Furthermore, $\varphi_i(\mathcal{L}/\det(\mathcal{L})\mathbb{Z}^n)$ is given by $B\mathbb{F}_{p_i}$, where B is a basis matrix for $\mathcal{L}$; thus, $\varphi_i(\mathcal{L}/\det(\mathcal{L})\mathbb{Z}^n)$ has dimension strictly less than n, since $\det(B) \equiv 0 \bmod p_i$. In particular, there are at most p_i^{n-1} elements in $\varphi_i(\mathcal{L}/\det(\mathcal{L})\mathbb{Z}^n)$, and hence there can be at most $p_1^{n-1} \cdots p_r^{n-1} = \det(\mathcal{L})^{n-1}$ elements $x \in \mathbb{Z}^n/\det(\mathcal{L})\mathbb{Z}^n$ that have, for all i, $x \equiv y_i \bmod p_i$, for some $y_i \in \mathcal{L}$. On the other hand, we also have that

$$|\mathcal{L}/\det(\mathcal{L})\mathbb{Z}^n| = \det(\det(\mathcal{L})\mathbb{Z}^n)/\det(\mathcal{L}) = \det(\mathcal{L})^n/\det(\mathcal{L}) = \det(\mathcal{L})^{n-1}$$

Since φ is a bijection, this proves that the elements of $\mathcal{L}/\det(\mathcal{L})\mathbb{Z}^n$ are the only elements of $\mathbb{Z}^n/\det(\mathcal{L})\mathbb{Z}^n$ that have, for all i, $x \equiv y_i \bmod p_i$, for some $y_i \in \mathcal{L}$. □

Proof of Correctness for Algorithm 4.1. If each sub-algorithm predicts correctly on its subspace, then lemma 4.3 guarantees that the prediction is correct. Therefore, if each sub-algorithm learns a hypothesis with error at most ϵ/r, then by a union bound, the error of the resulting hypothesis is at most ϵ. Note that $r \leq \log_2(\det(\mathcal{L})) \leq \log_2((m\sqrt{n})^n) = n\log_2(m\sqrt{n})$, so r is polynomial in n and $\log_2(m)$, ensuring that $1/(\epsilon/r)$ is polynomially related to $1/\epsilon$. As for the confidence, if each algorithm obtains a good hypothesis with probability at least $(1 - \delta/r)$, then they all succeed with probability at least $(1 - \delta/r)^r \geq 1 - \delta$. Finally, the most important observation is that if each example is corrupted by noise (either classification or malicious) with probability η, then each sub-algorithm receives examples that are corrupted with exactly the same probability. In particular, if the sub-algorithms can learn in the presence of noise with rate η, then so can the algorithm for learning full-dimensional square-free integer lattices. □

Conclusion

We have examined several connections between learning integer lattices and learning generalizations of parity in the presence of noise. We give an integer lattice learning problem that is equivalent to learning co-dimension 1 subspaces of $\mathbb{F}_p^n$ (a natural generalization of parity) in the presence of noise.

However, the restrictions present in the full-dimensional, square-free integer lattice problem are somewhat artificial. This suggests that learning arbitrary integer lattices in the presence of noise may in fact be significantly more difficult than learning parity, or more generally, co-dimension 1 subspaces of $\mathbb{F}_p^n$, in the presence of noise. Therefore, it may be fruitful to try to prove that lattices cannot be learned in the presence of noise, before attempting impossibility results for parity or other linear subspaces.

On the other hand, despite the apparent difficulty of learning integer lattices in the presence of noise, it is plausible that algorithms for learning parity in the presence of noise are more naturally stated in terms of learning lattices in the presence of noise. In fact, lattice algorithms such as those of [LLL82, Bab86], are routinely brought to bear on linear subspace problems in $\mathbb{F}_p^n$ (especially problems related to linear error-correcting codes), and perform much better than more naïve algorithms that work in $\mathbb{F}_p^n$ directly.

Finally, we should note that the reductions given here exploit very strong structural equivalences between integer lattices and linear subspaces of $\mathbb{F}_p^n$. As a consequence, the reductions carry over to other models of learning, such as the statistical query model. It would be of interest to study the effectiveness of these reductions in other such learning models.

References

[Bab86] L. Babai. On lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1993.

[CS93] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York, NY, USA, 2nd edition, 1993.

[HSW92] David Helmbold, Robert Sloan, and Manfred K. Warmuth. Learning integer lattices. *SIAM Journal on Computing*, 21(2):240–266, 1992.

[LLL82] Arjen K. Lenstra, H. W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[MW00] Daniele Micciancio and Bogdan Warinschi. A linear space algorithm for computing the hermite normal form. *Electronic Colloquium on Computational Complexity (ECCC)*, (074), 2000.