A UNIVERSAL DECODING ALGORITHM FOR LATTICE CODES

E. Viterbo **E. Biglieri**

Dipartimento di Elettronica • Politecnico • I-10129 Torino (Italy)

RÉSUMÉ

Le décodage des codes de réseau à n dimensions se fait en trouvant le vecteur le plus proche à un vecteur $\mathbf{z}$ donné dans l'espace euclidien $\mathbf{R}^n$. Le algorithmes de décodage trouvent des applications dans la quantification vectorielle et dans la démodulation de constellations de signaux multidimensionnels. Ici on montre un algorithme pour la solution du problème du décodage pour un code de réseau quelconque.

ABSTRACT

Decoding a d-dimensional lattice consists of finding the lattice point closest to any given vector $\mathbf{z}$ in the Euclidean space $\mathbf{R}^d$. Decoding algorithms find applications in vector quantization and in demodulation of multidimensional signal constellations. Here we describe an algorithm that solves the decoding problem for any lattice, irrespective of its particular algebraic structure.

1 Introduction

A lattice code is a finite subset of points of a lattice (or of a lattice translate) within a bounded region containing the origin, so that the energy of each signal is bounded. Lattice codes are used in vector quantization, where they provide highly structured codebooks with efficient encoding algorithms [10, p. 470 ff.], and in digital communications, where they generate signal constellations for high-rate transmission (see, e.g., [6] and the references therein). For sufficiently large signal-to-noise ratios, good constellations are usually carved from dense lattices, a selection prompted by De Buda's result that lattice codes asymptotically achieve Shannon's capacity bound [3].

A crucial procedure for both applications of lattice codes is their decoding. Given a d dimensional lattice Λ and a point $\mathbf{z}$ in the d-dimensional Euclidean space $\mathbf{R}^d$ in which Λ is embedded, decoding the lattice amounts to finding the point of Λ closest to $\mathbf{z}$. The problem here is to find this point without incurring the complexity of an exhaustive search. Practical algorithms which efficiently decode some well-known lattices that are attractive for applications ($A_n (n \geq 1)$, $D_n (n \geq 2)$, E_6, E_7, E_8, and their duals) are listed in [1]-[2, pp. 443 ff.]. Several Leech lattice decoders have been proposed with ever improving efficiency; a recent review of the subject can be found in [4].

The above algorithms are strictly dependent on the special structure of the lattice being decoded (e.g., its being a binary lattice [5]). Other algorithms [10, pp. 479–481] for general nearest neighbor encoding in vector quantization are valid for any unstructured codebook. They do not take full advantage of the lattice structure which is useful for large bit rates. The algorithm described in section 3 was first created as building block of a general Minkowski's basis reduction [9, 13]. We have adapted it to allow the decoding of any general lattice.

We may also observe here that, since any linear block code $\mathcal{C}$ over $\mathcal{Z}_q$ (the ring of integers modulo q) is in a sense equivalent to a sublattice of $\mathbf{Z}^n$ (see 'Construction A' in [2, Chapter 5]) any general decoding algorithm will also provide a quasi-Maximum Likelihood soft-decoding algorithm for $\mathcal{C}$.

2 Lattices

Definition 1 *Let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ be m linearly independent vectors of the d-dimensional Euclidean space $\mathbf{R}^d$ $(m \leq d)$. A lattice is the set Λ of vectors*

$$\lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \ldots + \lambda_m \mathbf{v}_m \qquad \lambda_1, \ldots, \lambda_m \in \mathbf{Z}$$

($\mathbf{Z}$ the relative integers). The set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ is called a basis of Λ.

If $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{id})$, $i = 1, \ldots, m$, we define the *generator matrix* of Λ as

$$M = \begin{pmatrix} v_{11} & \cdots & v_{1d} \\ \vdots & & \vdots \\ v_{m1} & \cdots & v_{md} \end{pmatrix},$$

so that we can simply write $\Lambda = \{\mathbf{u} = \mathbf{x}M : \mathbf{x} \in \mathbf{Z}^m\}$. Written in this form, we can view any lattice Λ as a transformed version of the *integer lattice* $\mathbf{Z}^m$: Λ is obtained by stretching $\mathbf{Z}^m$ along the coordinate axes and then rotating it around the origin. From now on we will consider only *full rank* lattices, i.e., those for which $m = d$ so that M is a non-singular square matrix. If M generates Λ, then any matrix of the form $M' = TM$, where T is an integer orthogonal matrix ($\det(T) = \pm 1$), is another generator matrix of Λ the same lattice: in fact T simply rotates $\mathbf{Z}^d$. T is also called an *integer unimodular matrix.*

The *fundamental parallelotope* of Λ is the set $\{\boldsymbol{\theta}\ M\}$, $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_d)$, $0 \le \theta_i < 1$. Its volume is equal to $|\det(M)|$, a number independent of the lattice basis and called the *determinant* of Λ. We indicate it with $\mathbf{d}(\Lambda)$.

The *Gram matrix* of Λ is defined as

$$A = MM^T = \begin{pmatrix} a_{11} & \cdots & a_{1d} \\ \vdots & & \vdots \\ a_{d1} & \cdots & a_{dd} \end{pmatrix}$$

Its elements are the Euclidean scalar products of pairs of vectors of the lattice basis, that is, $a_{ij} = (\mathbf{v}_i, \mathbf{v}_j) = \sum_k v_{ik} v_{jk}$. A is a symmetric positive definite matrix. Geometrically, the diagonal elements of A equal the squared norms of the basis vectors, while the other elements account for the inter-vector angles. It is immediate to see that $\det(A) = \mathbf{d}(\Lambda)^2$, so that the determinant of Λ is defined also when $m < d$. The quadratic form $Q(\mathbf{x}) = \mathbf{x}A\mathbf{x}^T = \sum_{i,j} a_{ij} x_i x_j$, $\mathbf{x} \in \mathbf{R}^d$, is positive definite, with *discriminant* $\delta(Q) = \det(A)$. If we are only interested in the metric properties of a lattice, we can equivalently work on the basis vectors or on the form Q restricted to $\mathbf{Z}^d$. For example, the minimum squared Euclidean distance between any two points of Λ equals the minimum of $Q(\mathbf{x})$ for $\mathbf{x} \in \mathbf{Z}^d \backslash \{0\}$.

3 The decoding algorithm

As a preliminary to the decoding problem, we first describe an algorithm for computing the shortest nonzero vector of a lattice. The basic idea in both cases is to restrict the search to a finite number of lattice points which lie within a bounded region. We generally start with a large region and, whenever a shorter vector is found, the region is consequently restricted.

3.1 Shortest nonzero vector in a lattice

The task of determining nonzero vectors of Λ with shortest length was first considered by Gauss and later by Minkowski in his 'Geometry of Numbers'. Minkowski's Fundamental Theorem provides an upper bound to the length of such vector [8]. The first application which tackled the computational aspect of the problem can be found in the study of the lattice structure of pseudo-random numbers generated by the linear congruential method [12].

One of the first algorithms proposed is described in [7]. Here, the search region is defined based on the dual of Λ. A preliminary base reduction can restrict the size of the starting region, but the search becomes prohibitively complex when the lattice dimensionality grows above a certain threshold (around 10).

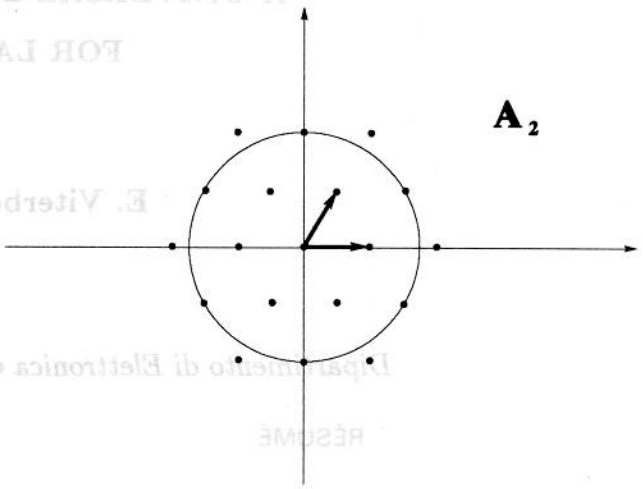

Figure 1: The two-dimensional hexagonal lattice

A substantial improvement was introduced by Pohst in [13]. and further analyzed in [9]. We briefly illustrate this algorithm here, and provide some further insight through its geometrical interpretation. Consider a vector $\mathbf{u} \in \mathbf{R}^d$, and let $\|\mathbf{u}\| = \sqrt{\mathbf{u}\mathbf{u}^T}$ denote its Euclidean norm. A *ball* of radius $\sqrt{C}$ is defined by the inequality

$$\|\mathbf{u}\|^2 \le C. \tag{1}$$

If $\mathbf{u}$ is a lattice point, then it can be written in the form $\mathbf{u} = \mathbf{x}M$ for some integer vector $\mathbf{x} \in \mathbf{Z}^d$, and lies inside the ball if

$$\mathbf{x}MM^T\mathbf{x}^T = \mathbf{x}A\mathbf{x}^T = \sum_{i=1}^{d}\sum_{j=1}^{d} a_{ij} x_i x_j \le C \tag{2}$$

If we let the vector $\mathbf{x}$ take on real values, then (2) is the equation of an ellipsoid (Figs. 1 and 2), the lengths of whose semi-axes are directly related to eigenvalues λ_i of the matrix A.

Cholesky's factorization yields $A = R^T R$, where R is an upper triangular matrix with elements $r_{ij} = 0$ for $2 \le i \le j \le d-1$ and $r_{ii} = \sqrt{\lambda_i}$. Then

$$\mathbf{x}A\mathbf{x}^T = \|R\mathbf{x}^T\|^2 = \sum_{i=1}^{d} \left(r_{ii} x_i + \sum_{j=i+1}^{d} r_{ij} x_j \right)^2 \le C. \tag{3}$$

Substituting $q_{ii} = r_{ii}^2$ for $i = 1, \ldots, d$ and $q_{ij} = r_{ij}/r_{ii}$ for $i = 1, \ldots, d$, $j = i+1, \ldots, d$, we can write

$$Q(\mathbf{x}) = \sum_{i=1}^{d} q_{ii} \left(x_i + \sum_{j=i+1}^{d} q_{ij} x_j \right)^2 \le C. \tag{4}$$

The canonical form of this ellipsoid

$$\sum_{i=1}^{d} q_{ii} X_i^2 \le C \tag{5}$$

is obtained by using a suitable coordinate tranformation. The value of each component of a point inside the ellipsoid can be bounded in the new coordinate system $\{X_1, \ldots, X_{d-1}, X_d\}$ based on the axes of the ellipsoid.

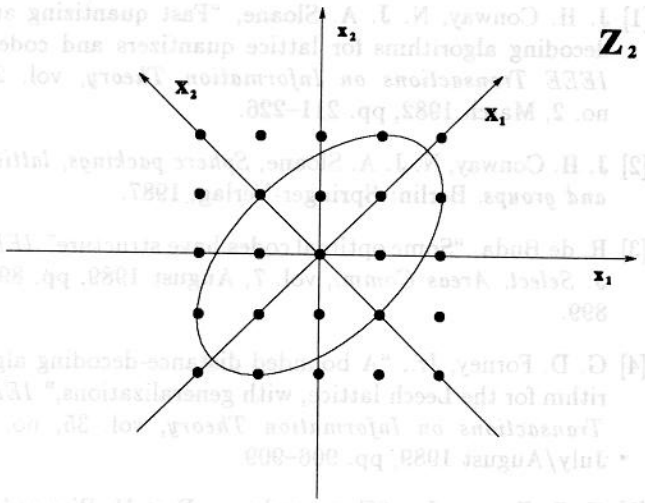

Figure 2: The integer lattice with the transformed region

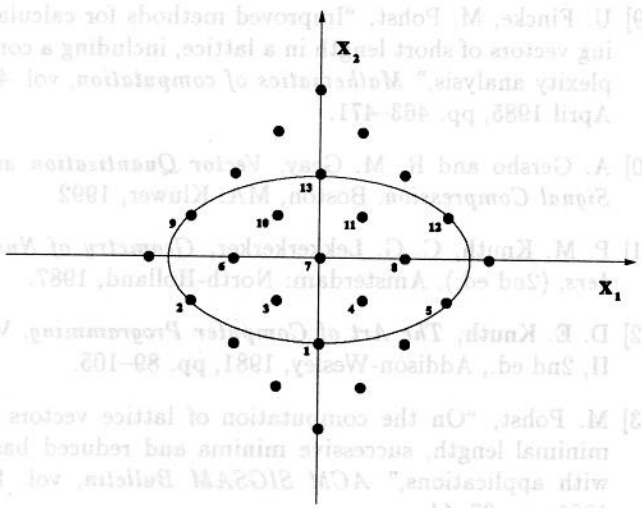

Figure 3: The integer lattice in the new coordinate system. The points inside the ellipse are numbered according to the order in which they are tested by the algorithm.

The search algorithm procedes very much like a mixed radix counter, with the addition that the bounds change whenever there is a carry operation from one digit to the next. (Figure 3 shows how the algorithm works. It scans exactly all the points of $\mathbf{Z}^2$ inside the ellipse in the order indicated by the numbers).

If a vector $\mathbf{u} = \mathbf{x}M$ for some $\mathbf{x} \in \mathbf{Z}^d$, such that $\|\mathbf{u}\|^2 < C$, is found, then we are able to reduce the radius of the ball. We substitute $\|\mathbf{u}\|^2$ for C, we update all the bounds, and we keep on searching in the smaller ball without restarting from the beginning.

The great advantage of this method over [7] lies in the fact that we never test vectors with a norm greater than the given radius. Every vector tested requires the computation of its norm, which entails d multiplications and $d-1$ additions. The increase in number of operations needed to update the bounds is largely compensated for by the enormous reduction in the number of vectors tested.

As a byproduct of this algorithm we may obtain the *kissing number* of Λ, that is, the number of lattice points at the minimum distance from the origin.

3.2 Closest lattice point

Here we want to solve the problem

$$\min_{\mathbf{u}\in\Lambda} \|\mathbf{z}-\mathbf{u}\| = \min_{\mathbf{w}\in\mathbf{z}-\Lambda} \|\mathbf{w}\|. \tag{6}$$

We write $\mathbf{u} = \mathbf{x}M$ with $\mathbf{x} \in \mathbf{Z}^d$, $\mathbf{z} = \boldsymbol{\zeta} M$ with $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_d)$, and $\mathbf{w} = \boldsymbol{\xi} M$ with $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_d)$, where $\boldsymbol{\zeta}$ and $\boldsymbol{\xi}$ are real vectors. Then we have $\mathbf{w} = \sum_{i=1}^{d} \xi_i \mathbf{v_i}$ where $\xi_i = \zeta_i - x_i$, $i = 1, \ldots, d$. We now have to find the shortest nonzero vector of the translated lattice $\mathbf{z} - \Lambda$. As before, we construct a ball of radius $\sqrt{C}$ centered at $\mathbf{z}$ and we test all the lattice points that are inside.

Some additional comments on the choice of $\sqrt{C}$ are appropriate here. The furthest point of $\mathbf{R}^d$ from a point of Λ is called a *deep hole* of the lattice. The *covering radius* R of Λ is the smallest distance of a lattice point from a deep hole. If the covering radius of the lattice is known, then we take it as the starting value for $\sqrt{C}$: otherwise we may use Roger's upper bound to the covering radius [11, p. 241].

In a practical application the radius could be adaptively adjusted according to the noise level in the following way. If no lattice point is detected inside the ball, the radius must be increased and an erasure can be indicated to the higher levels. On the contrary, when the distance of the received point from the lattice point $\|\mathbf{w}\|$ is small, then the radius can be decreased.

4 Soft-decoding of the ternary (12, 6, 6) Golay code

As an example of application of the algorithm described before, we study its complexity when used to soft-decode the ternary (12, 6, 6) Golay code. The generator matrix of the corresponding lattice is

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 2 & 1 & 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 2 & 2 & 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 2 & 2 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 1 & 2 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix} \quad (7)$$

which corresponds to periodically repeating in each direction the code vectors contained in a 12-dimensional cube of edge 3. The covering radius of this lattice is not known and Roger's upper bound gives $R = 2.48$ (we conjecture $R = 2$, a value that has never been exceeded in our computations). The simulation was performed by generating a sequence of vectors $\mathbf{z}$ in the form $\mathbf{x} + \mathbf{n}$, where $\mathbf{n}$ is a zero-mean Gaussian vector with independent, identically distributed components. According to simulation, the average numbers of additions, multiplications and square roots per codeword are about 4900, 3800 and 350. For a complete search through code book we would need 16767 additions and 8748 multiplications, as well as the complete storage of all the 729 ternary codewords. Although square roots are usually considered as lengthy operations, in our case they can be performed with a reduced precision, since they are followed by a floor or ceiling function. Finally we observe that the decoding algorithm is not exactly Maximum Likelihood since it can decode a lattice point which is not a code vector. In general this difference is only noticeable for very low signal to noise ratios and for small values of q. To better approximate the ML decoding various strategies can be adopted

1. reduce modulo 3 the lattice point coordinates produced by the decoder;
2. during the search in the sphere, discard the points which are not code vectors; if no code vector is found declare an erasure or use strategy 3;
3. take a hard decision on the information part of the received vector.

Note that when any of the above strategies is adopted, the uniform error property is no longer valid since the decoder is performing some non linear operation.

In order to obtain ML decoding, we are currently investigating a noise compression technique which forces large received vectors to lie on the surface of the smallest sphere containing all the code points.

5 Conclusions

The algorithm we have presented shows the advantages offered by the continuous structure of the space in which lattices are embedded. This algorithm was also tested as a Leech lattice decoder and the total number of operations was about 885,000, where additions, multiplications, and square roots were in the same proportions as in the previous example. This result may look discouraging when compared with 8000 operations required by the fastest Leech decoding algorithm known [4]. However, it should be kept in mind that the generality of our procedure enables one to decode lattices for which no *ad hoc* algorithm is known.

References

[1] J. H. Conway, N. J. A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes" *IEEE Transactions on Information Theory*, vol. 28, no. 2, March 1982, pp. 211–226.

[2] J. H. Conway, N. J. A. Sloane, *Sphere packings, lattices and groups*. Berlin: Springer-Verlag, 1987.

[3] R. de Buda, "Some optimal codes have structure" *IEEE J. Select. Areas Comm.*, vol. 7, August 1989, pp. 893–899.

[4] G. D. Forney, Jr., "A bounded distance-decoding algorithm for the Leech lattice, with generalizations," *IEEE Transactions on Information Theory*, vol. 35, no. 4, July/August 1989, pp. 906–909.

[5] G. D. Forney, Jr., "Coset codes — Part II: Binary lattices and related codes," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1152–1187, September 1988.

[6] G. D. Forney, Jr., "Multidimensional constellations — Part II: Voronoi constellations," *IEEE J. Select. Areas Comm.*, vol. 7, no. 4, August 1989, pp. 941–958.

[7] U. Dieter, "How to calculate shortest vectors in a lattice," *Mathematics of Computation*, vol. 29, July 1975, pp. 827–833.

[8] P. Erdös, P. M. Gruber, J. Hammer, *Lattice points*. Essex, England: Longman Scientific & Technical, 1989.

[9] U. Fincke, M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of computation*, vol. 44, April 1985, pp. 463–471.

[10] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.

[11] P. M. Knuth, C. G. Lekkerkerker, *Geometry of Numbers*, (2nd ed.), Amsterdam: North-Holland, 1987.

[12] D. E. Knuth, *The Art of Computer Programming*, Vol II, 2nd ed., Addison-Wesley, 1981, pp. 89–105.

[13] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications," *ACM SIGSAM Bulletin*, vol. 15, 1981, pp. 37–44.