1 Introduction

Roughly speaking, a lattice is a periodic arrangement of points in the *n*-dimensional Euclidean space.[1] It reflects the "geometry of numbers" – in the words of the late nineteenth century mathematician Hermann Minkowski. Except for the one-dimensional case (where all lattices are equivalent up to scaling), there are infinitely many shapes of lattices in each dimension. Some of them are better than others.

Good lattices form effective structures for various geometric and coding problems. Crystallographers look for symmetries in three-dimensional lattices, and relate them to the physical properties of common crystals. A mathematician's classical problem is to pack high-dimensional spheres – or cover space with such spheres – where their centers form a lattice. The communication engineer and the information theorist are interested in using lattices for quantization and modulation, i.e., as a means for lossy compression (source coding) and noise immunity (channel coding). Although these problems seem different, they are in fact closely related.

The effectiveness of good lattices – as well as the complexity of describing or using them for coding – increases with the spatial dimension. Such lattices tend to be "perfect" in all aspects as the dimension goes to infinity. But what does "goodness" mean in dimensions 2, 3, 4, . . .?

In two dimensions, the *hexagonal lattice* is famous for the honeycomb shape of its Voronoi cells. The centers of the billiard (pool) balls in Figure 1.1 fall on a hexagonal lattice, which forms the tightest packing in two dimensions. The same hexagonal lattice defines a configuration for deploying cellular base stations that maximizes the coverage area per base station.

Interestingly, however, for higher dimensions the problems of packing and covering are *not* equivalent. In Figure 1.2, the centers of the oranges fall on the face-centered cubic (FCC) lattice, which is the best known sphere packing in three dimensions. In contrast, the best deployment of cellular base stations in a skyscraper (which maximizes their *three-dimensional* coverage) is over a body-centered cubic (BCC) lattice, illustrated in Figure 1.3.

[1] See the Wikipedia disambiguation page for other meanings of the word "lattice": in art and design, music, math and science.

Figure 1.1 Billiard (pool) balls packed in a triangle, for an initial game position.

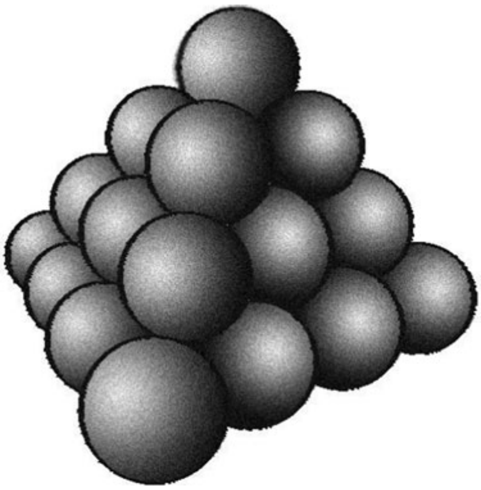

Figure 1.2 Packing oranges in a pile: each row is half-diameter shifted with respect to the previous row to reduce the unused volume. Similarly, each layer is staggered to fill the holes in the layer below it. The centers of the oranges form a lattice known as a face-centered cubic (FCC) lattice.

Which is the "best" lattice in each dimension is a question we shall not address; issues of efficient design and coding complexity of lattices are not at the focus of this book either. Instead, we characterize the performance of a lattice code by its thickness (relative excess coverage) and density (relative packed volume), and by the more communication-oriented figures of merit of normalized second moment (NSM) for quantization, and normalized volume to noise ratio (NVNR) for modulation. We define these quantities in detail in Chapter 3, and use them in Chapters 4–9 to evaluate lattice codes for the basic *point-to-point* source and channel coding problems. As we shall see, high-dimensional lattice codes can close

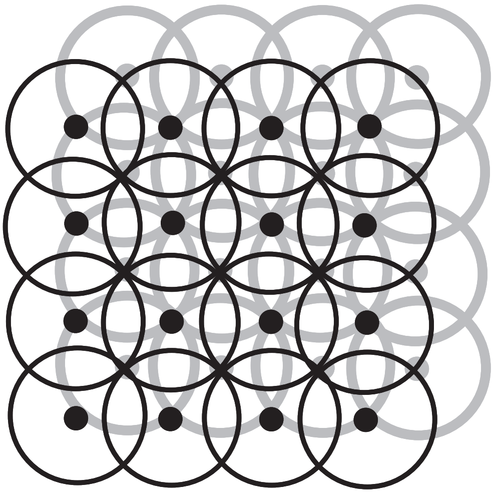

Figure 1.3 Three-dimensional sphere covering with a BCC lattice, describing the best deployment of cellular base stations in a skyscraper. The solid line shows even layers; the gray line shows odd layers. Compare the staggering pattern with that of the pile of oranges in Figure 1.2.

the gap to the information theoretic limits of communication: the capacity and rate-distortion function, quantities introduced by Shannon in his seminal 1948 paper [240], and further refined during the 1950s and 1960s.

The 1970s and 1980s saw the blooming of network information theory. Remarkably, some of the fundamental network problems were successfully solved using Shannon's information measures and *random coding* techniques, now with the additional variant of random binning. Simple examples of such network setups are *side-information* problems: the Slepian–Wolf and Wyner–Ziv source coding problem, and the Gelfand–Pinsker "dirty-paper" channel coding problem. The lattice framework provides a *structured coding* solution for these problems, based on a nested pair of lattices. This nested lattice configuration calls for new composite figures of merit: one component lattice should be a good channel code (have a low NVNR), while the other component lattice should be a good quantizer (have a low NSM). For joint source-channel coding problems, lattices with a good NSM-NVNR product are desired. We shall develop these notions in Chapters 10 and 11.

The curious reader may still wonder why we need a book about lattices in information theory. After all, Shannon's probabilistic measures and random coding techniques characterize well the limits of capacity (channel coding) and compression (source coding), and they also allow the study of source and channel networks [53, 64]. From the practical world side, communication theory provides ways to combine modulation with "algebraic" codes and approach the Shannon limits.

All this is true, yet between the theoretical and the constructive points of view something gets lost. Both the probabilistic and the algebraic approaches somewhat

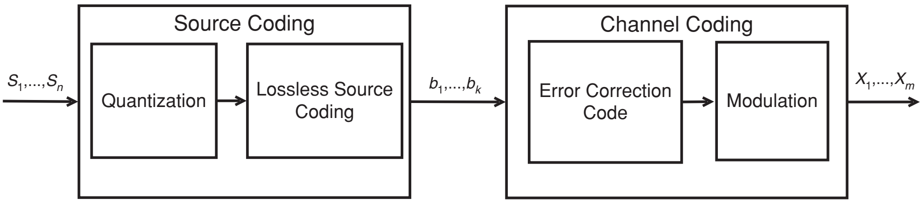

Figure 1.4 Source coding followed by channel coding. For an analog source and channel, the combined system maps a point in $\mathbb{R}^n$ (a source vector) to a point in $\mathbb{R}^m$ (a channel input vector). The ratio m/n is known as the "bandwidth-expansion factor."

hide the interplay between analog signals like sound or noise (created by nature) and digital modulation signals (created by man). Lattices are discrete entities in the analog world, and as such they bridge nicely the gap between the two worlds. At large dimensions, good lattices mimic the behavior of Shannon's random codes. For small dimensions, they represent an elegant combination of modulation and digital coding. As a whole, lattices provide a unified framework to study communication and information theory in an insightful and inspiring way.

Recent developments in the area of network information theory (mostly from the 2000s) have added a new chapter to the story of lattice codes. In some setups, structured codes are potentially *performance-wise better* than the traditional random coding schemes! And as Chapter 12 shows, the natural candidates to achieve the benefit of structure in Gaussian networks are, again, lattice codes.

1.1 Source and channel coding

Let us describe briefly how lattices fit into the framework of digital communication and classical information theory.

By Shannon's *separation principle*, transmission of an information source over a noisy channel is split into two stages: *source coding*, where the source is mapped into bits, and *channel coding*, where the digital representation of the source is mapped into a channel input signal. These two stages, which we describe in detail below, are illustrated in Figure 1.4.

The *source coding* (or compression) problem deals with compact digital representation of source signals. In *lossless* compression, our goal is to *remove redundancy* due to asymmetry in the frequency of appearance of source values, or to "memory" in the source. In this case, the source signal is available already in a digital form, say, as a sequence of binary symbols. And the task is to map n "redundant" source bits $\mathbf{s} = s_1, \ldots, s_n$ into $k = k(\mathbf{s})$ code bits, where $k < n$.[2]

[2] We would like k to be smaller than n for most source vectors (or for the most likely ones) in order to compress; but not too small, so the mapping would be invertible for (almost) all source vectors, for lossless reproduction.

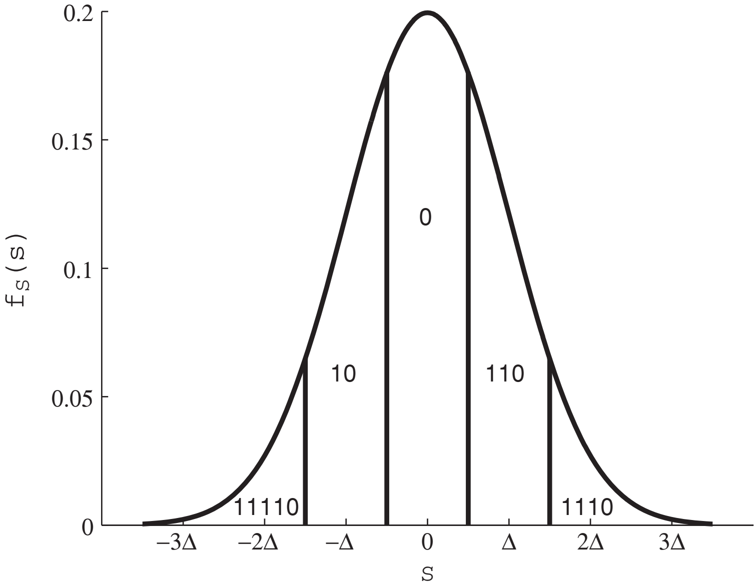

Figure 1.5 Scalar uniform quantization of a Gaussian source, followed by variable-length coding, i.e., $n = 1$ and k is varying. Each quantization level represents a range of source values.

In *lossy* compression, the source is usually *continuous* in nature: an analog representation of speech, sound, picture or video signal. Digitizing an analog signal consists first of converting it into a *discrete* form (both in time and in amplitude), and then coding it in the discrete alphabet domain. In discrete time the source is again a vector $\mathbf{s} = s_1, \ldots, s_n$, representing n consecutive source samples. After the vector $\mathbf{s}$ is encoded into a k-bit codeword, it is decoded and reconstructed as $\hat{\mathbf{s}} = \hat{s}_1, \ldots, \hat{s}_n$. The overall operation of mapping $\mathbf{s}$ to $\hat{\mathbf{s}}$ is called *quantization*, and the image (for a fixed k, the set of all 2^k possible reconstruction vectors $\hat{\mathbf{s}}$ in $\mathbb{R}^n$) is the quantization *codebook*.

A *lattice quantizer* codebook consists of points from an n-dimensional lattice. The codebook can be a truncated version (of size 2^k) of the lattice, or the whole lattice (with a variable codeword length $k = k(\hat{\mathbf{s}})$). We would like to make the bit rate $R = k/n$ (or the average coding rate $R = \bar{k}/n$) as *small* as possible, subject to a constraint on the reconstruction fidelity. Figure 1.5 shows the case of a scalar ($n = 1$) lattice quantizer with a variable code length $k(\hat{\mathbf{s}})$.

Channel coding deals with transmitting or storing information over a noisy channel or on a storage device. Our goal here is to *add redundancy* to the transmitted signal, to make it distinguishable from the noise. The channel input alphabet may be *discrete*, say, binary. In this case, transmission amounts to mapping k bits of information into n "redundant" code bits, where $n > k$.

The most common communication links are, however, over *continuous* media: telephone lines, cables or radio waves. The *baseband* channel representation is in

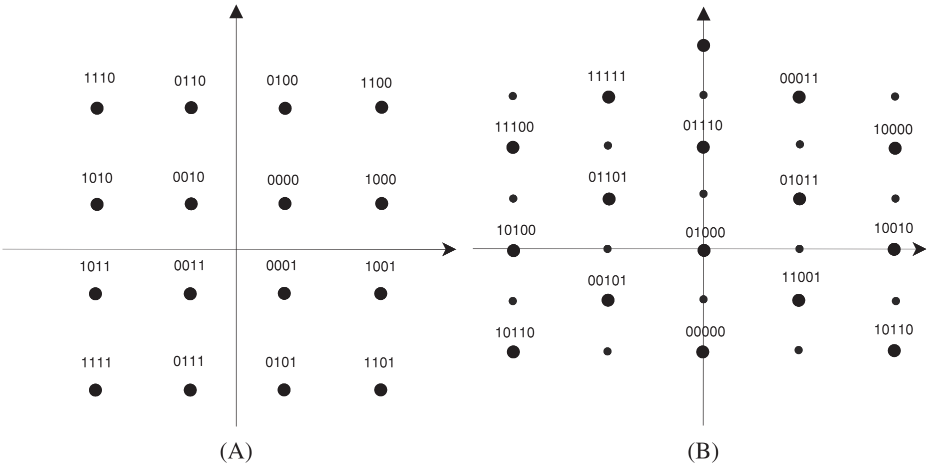

Figure 1.6 Two-dimensional finite lattice constellations, consisting of 16 points ($k = 4$). (A) A simple square constellation, representing uncoded quadrature-amplitude modulation (QAM); here $n' = k = 4$. (B) A hexagonal lattice constellation, represented as a mapping of redundant binary vectors of length $n' = 5$ into a rectangular constellation.

discrete time, so the channel input is a vector $\mathbf{x} = x_1, \ldots, x_n$. Coding over such a channel turns out to be in many ways the *dual* of encoding an analog source. It consists of two stages: an *error-correction coding* stage, where redundancy is added in the discrete alphabet domain (e.g., by converting k information bits to $n' > k$ code bits); and a *modulation* stage, where the digital codeword is mapped into the vector $\mathbf{x}$. The overall encoder mapping is thus of a k-bit information vector into a point in $\mathbb{R}^n$ (representing n consecutive channel inputs). The set of all 2^k possible input vectors $\mathbf{x}$ is called a codebook or a *constellation*.

A *lattice constellation* is a truncated version (of size 2^k) of an n-dimensional lattice. We would like to make the coding rate $R = k/n$ – which is now the (usually fixed) number of transmitted information bits per channel input – as *large* as possible, subject to a constraint on the probability of decoding error. See two examples of two-dimensional lattice constellations in Figure 1.6.

One benefit of the lattice coding framework that we can immediately recognize is that coding and modulation (or quantization) are combined as a *single entity*; a lattice code directly maps digital information (say, an index) into a vector in $\mathbb{R}^n$, and vice versa.

1.2 The information theoretic view

Information theory characterizes the ultimate performance limits of source and channel coding, as the code block length n goes to infinity.

In the channel coding case, the coding rate R is upper bounded (for a vanishing error probability) by the *Shannon capacity* C of the channel. The quantity C (associated with a memoryless channel with a transition distribution $p(y|x)$) is calculated by maximizing the mutual information (a functional of $p(x)$ and $p(y|x)$) over the input distribution $p(x)$. The maximizing input distribution $p^*(x)$ is used to prove the achievability of C: a set of $\approx 2^{nC}$ codewords is generated *randomly* and independently with an i.i.d. distribution $p^*(x)$; a *random coding* argument is then used to show that based on the channel output, the decoder can guess the correct transmitted codeword with a high probability as $n \to \infty$.

We see that *à la* Shannon, good codewords look like *realizations of random noise*. In the case of a binary-symmetric channel, the code generating noise consists of equally likely 0/1 bits. In the quadratic-Gaussian case, the code should be generated by a *white-Gaussian noise* (WGN).

Rate-distortion theory uses similar ideas to establish the ultimate performance limits of lossy source coding [18]. The Shannon *rate-distortion function* $R(D)$ lower bounds the coding rate R of any lossy compression scheme with distortion level of at most D (under some given distortion measure). And similarly to the channel coding case, computation of $R(D)$ induces an optimal reconstruction distribution, which is used to generate a good random codebook: independent realizations of a Bernoulli(1/2) sequence compose the codewords for a binary-symmetric source under Hamming distortion, while independent realizations of WGN compose the codewords for a white-Gaussian source under mean-squared distortion.

The fact that good codewords look like white noise is intriguing. Intuitively, one would expect the symbols of a codeword to be *dependent*, to distinguish them from the channel noise. This has made the random coding idea, on the one hand, a source of inspiration for many since Shannon presented his landmark theory in 1948. On the other hand, it sets a challenge for finding more *structured* ways to approach the information theoretic limits, ways in which the dependence between the code symbols is more explicit. Can noise be realized in a structured way?

1.3 Structured codes

The Hamming code – mentioned already in Shannon's 1948 paper – was the early bird of the structured coding approach. It was followed by the breakthrough of algebraic coding theory in the 1950s and 1960s [21]. The implication was that, in fact, a good collection of random-like bits can be constructed as an additive group in the binary modulo-2 space. These *linear codes* take various forms, such as Reed–Muller, BCH and, more recently, LDPC, turbo and polar codes, and they also have extensions to non-binary (Reed–Solomon) codes and convolutional (trellis) codes.

Common to all these codes is that for a random message, the resulting n-length codeword is indeed roughly uniformly distributed over the n-dimensional binary space. That is, each code bit takes the values 0 and 1 with equal probability; furthermore, small subsets of code bits are roughly independent.

The extension of this concept to continuous signals is however less obvious: can a code mimic Gaussian noise in a structured way? A first step towards this goal is provided by Shannon's *asymptotic equipartition property* (AEP). In a high dimension n, the *typical set* of WGN of variance σ^2 is a spherical shell of radius $\approx \sqrt{n\sigma^2}$. Thus, the codewords of a good code are roughly uniformly distributed over such a spherical shell.

The concept of *geometrically uniform codes* (GUC) [86] suggests a deterministic characterization for a "uniform-looking" code: every codeword should have the same *distance spectrum* to its neighboring codewords. This concept captures the desired property of a good Euclidean code, in both the block and the convolutional (*trellis*) coding frameworks.

Due to their periodic and linear structure, lattices are natural candidates for *unbounded* GUCs. For example, the commonly used QAM constellation shown in Figure 1.6(A) is a truncated version of the *square lattice*, while the more "random-like" set of two-dimensional codewords shown in Figure 1.6(B) is a truncated version of the *hexagonal lattice*. Moreover, the code designer can shape the borders of these constellations to be more round, for example, by truncating them into a circle or into a coarser hexagonal cell. And as the dimension gets high, lattices which are truncated into a "good" coarse lattice cell become closer to a randomly generated Gaussian codebook.

1.4 Preview

We shall get to the exciting applications mentioned earlier after building up some necessary background. The book starts by introducing lattices in Chapter 2, and the notions of lattice goodness in Chapter 3. Chapter 4 introduces two central players in our framework: dithering, which is a means to randomize a lattice code, and Wiener estimation, which is a means to reduce the quantization or channel noise. The importance of these techniques will be revealed gradually throughout the book.

Equipped with these notions and techniques, we consider variable-rate ("entropy-coded") dithered quantization (ECDQ) using an *unbounded* lattice in Chapter 5. In particular, we shall see how the NSM characterizes the redundancy of the ECDQ above Shannon's rate-distortion function. The reader who is interested primarily in channel coding may skip Chapter 5, and continue directly to modulation with an unbounded lattice constellation in Chapter 6.[3] This chapter shows how the NVNR determines the gap from capacity of a lattice constellation. It also describes variable-rate dithered modulation, which is the channel coding counterpart of ECDQ.

[3] Sections which are optional reading for the flow of the book are denoted by an asterisk.

Before moving to more advanced coding setups, we stop to examine the existence of asymptotically good lattices in Chapter 7. In Chapter 8 we define nested lattices, and *finite* Voronoi-shaped codebooks taken from a lattice. These notions form in Chapter 9 the basis for Voronoi modulation, which achieves the capacity of a power-constrained AWGN channel, and for Voronoi quantization, which achieves the quadratic-Gaussian rate-distortion function. In both these solutions, dither and Wiener estimation play crucial roles.

A small step takes us from the point-to-point communication setups above to side-information problems in Chapter 10. We shall construct lattice code solutions for the Wyner–Ziv problem (source coding with side information at the decoder) and the "dirty-paper" problem (channel coding with side information at the encoder). These lattice coding schemes serve as building blocks for common multi-terminal communication problems: encoding of distributed sources and broadcast channels. Before moving to more general networks, we examine in Chapter 11 a lattice-based joint source-channel coding technique, called modulo-lattice modulation (MLM). A combination of MLM and prediction leads to "analog matching" of sources and channels with mismatched spectra, and to "bandwidth conversion." Chapter 12 extends the discussion on multi-terminal problems to general Gaussian networks. There we shall see that when side information is distributed among several nodes of the network, lattice codes are not only attractive complexity-wise, but sometimes they have better performance than traditional random coding and binning techniques.

Chapter 13 complements the discussion of asymptotically good lattice codes in Chapter 7 by examining their error exponents. As for capacity, good lattice codes turn out to be optimal also in terms of this more refined aspect.

Information theory is not a critical prerequisite for reading this book, but (starting from Chapter 5) we use information measures, such as entropy, mutual information and capacity, to assess system performance. To keep the book self-contained, the Appendix includes elementary background in information theory, as well as some other complementary material.

As mentioned above, dithering and Wiener estimation are central concepts in the lattice coding framework. The question of where and in what sense they are necessary will follow our discussion throughout the book.

What's *not* in the book?

The writer has the freedom to focus on his favorite subject. Naturally (in the case of this writer) the book takes an information theoretic flavor, with less emphasis on coding theoretic aspects. For algebra of lattices, and for specific constructions of lattices and coded-modulation schemes from error-correcting codes, the reader is referred to the comprehensive book of Conway and Sloane [49], and to the excellent class notes of Forney [81] and Calderbank [28].

Encoding and decoding complexity is a topic of theoretical as well as practical importance, although traditionally neglected by information theory. A good introduction to the subject can be found in the survey paper of Agrell *et al.* [3]. The vast literature on MIMO communication contains numerous publications about the design of linear coded-modulation schemes and efficient lattice decoding algorithms.

In the fight between a timely manuscript and time of publication, some topics which are natural to the spirit of the book were left out. One such topic is the extension to *colored*-Gaussian sources and channels; see, for example, [211, 288, 291]. Another topic is the emerging area of lattice wiretap codes; see, for example, the survey paper by Liang *et al.* [156] and other recent work [118, 168]. Hopefully these topics will find their way to a later edition of the book.

Finally, since the late 1990s lattice-based cryptography has been a major area of research in computer science. Its connection to lattice codes for communication is yet to be explored; see the book by Micciancio and Goldwasser [186], and the survey by Micciancio and Regev [188].