1 Motivation

1.1 The importance of compression

It is easy to recognize the importance of data compression technology by observing the way it already pervades our daily lives. For instance, we currently have more than a billion users [1] of digital cameras that employ JPEG image compression, and a comparable number of users of portable audio players that use compression formats such as MP3, AAC, and WMA. Users of video cameras, DVD players, digital cable or satellite TV, hear about MPEG-2, MPEG-4, and H.264/AVC. In each case, the acronym is used to identify the type of compression. While many people do not know what exactly compression means or how it works, they have to learn some basic facts about it in order to properly use their devices, or to make purchase decisions.

Compression's usefulness is not limited to multimedia. An increasingly important fraction of the world's economy is in the transmission, storage, and processing of all types of digital information. As Negroponte [2] succinctly put it, economic value is indeed moving "from atoms to bits." While it is true that many constraints from the physical world do not affect this "digital economy," we cannot forget that, due to the huge volumes of data, there has to be a large physical infrastructure for data transmission, processing, and storage. Thus, just as in the traditional economy it is very important to consider the efficiency of transportation, space, and material usage, the efficiency in the representation of digital information also has great economic importance.

This efficiency is the subject of this book. Data compression encompasses the theory and practical techniques that are used for representing digital information in its most efficient format, as measured (mostly) by the number of bits used for storage or telecommunication (bandwidth). Our objective is to present, in an introductory text, all the important ideas required to understand how current compression methods work, and how to design new ones.

A common misconception regarding compression is that, if the costs of storage and bandwidth fall exponentially, compression should eventually cease to be useful. To see why this is not true, one should first note that some assumptions about costs are not universal. For example, while costs of digital storage can indeed fall exponentially, wireless telecommunications costs are constrained by the fact that shared radio spectrum is a resource that is definitely limited, land line communications may need large new investments, etc. Second, it misses the fact that the value of compression is unevenly

distributed according to applications, type of user, and in time. For instance, compression is commonly essential to enable the early adoption of new technologies – like it was for digital photography – which would initially be too expensive without it. After a while, it becomes less important, but since the infrastructure to use it is already in place, there is little incentive to stop using it. Furthermore, there is the aspect of relative cost. While even cheap digital cameras may already have more memory than will ever be needed by their owners, photo storage is still a very important part of the operational costs for companies that store the photos and videos of millions of users. Finally, it also ignores the fact that the costs for generating new data, in large amounts, can also decrease exponentially, and we are just beginning to observe an explosion not only in the volume of data, but also in the number and capabilities of the devices that create new data (cf. Section 1.4 and reference [1]).

In conclusion, we expect the importance of compression to keep increasing, especially as its use moves from current types of data to new applications involving much larger amounts of information.

1.2 Data types

Before presenting the basics of the compression process, it is interesting to consider that the data to be compressed can be divided into two main classes, with different properties.

1.2.1 Symbolic information

We can use the word *text* broadly to describe data that is typically arranged in a sequence of arbitrary symbols (or *characters*), from a predefined *alphabet* or *script* (writing system) of a given size. For example, the most common system for English text is the set of 8-bit ASCII characters, which include all letters from the Latin script, plus some extra symbols. It is being replaced by 16-bit UNICODE characters, which include all the important scripts currently used, and a larger set of special symbols.

Normally we cannot exploit the numerical value of the digital representation of symbolic information, since the identification of the symbols, and ultimately their meaning, depend on the convention being used, and the character's context. This means that the compression of symbolic data normally has to be *lossless*, i.e., the information that is recovered after decompression has to be identical to the original. This is usually what is referred to as *text compression* or *data compression.*

Even when all information must be preserved, savings can be obtained by removing a form of *redundancy* from the representation. Normally, we do not refer to compression as the simplest choices of more economical representations, such as converting text that is known to be English from 16-bit UNICODE to 8-bit ASCII. Instead, we refer to compression as the techniques that exploit the fact that some symbols, or sequences

of symbols, are much more commonly used than others. As we explain later, it is more efficient to use a smaller number of bits to represent the most common characters (which necessarily requires using a larger number for the less common). So, in its simplest definition, lossless compression is equivalent simply to reducing "wasted space." Of course, for creating effective compression systems, we need a rigorous mathematical framework to define what exactly "wasted" means. This is the subject of *information theory* that is covered from Chapter 2.

1.2.2 Numerical information

The second important type of data corresponds to information obtained by measuring some physical quantity. For instance, audio data is created by measuring sound intensity in time; images are formed by measuring light intensity in a rectangular area; etc. For convenience, physical quantities are commonly considered to be real numbers, i.e., with infinite precision. However, since any actual measurement has limited precision and accuracy, it is natural to consider how much of the measurement data needs to be preserved. In this case, compression savings are obtained not only by eliminating redundancy, but also by removing data that we know are *irrelevant* to our application. For instance, if we want to record a person's body temperature, it is clear that we only need to save data in a quite limited range, and up to a certain precision. This type of compression is called *lossy* because some information—the part deemed irrelevant—is discarded, and afterwards the redundancy of the remaining data is reduced.

Even though just the process of discarding information is not by itself compression, it is such a fundamental component that we traditionally call this combination *lossy compression.* Methods that integrate the two steps are much more efficient in keeping the essential information than those that do not. For instance, the reader may already have observed that popular lossy media compression methods such as the JPEG image compression, or MP3 audio compression, can achieve one or two orders of magnitude in size reduction, with very little loss in perceptual quality.

It is also interesting to note that measurement data commonly produce relatively much larger amounts of data than text. For instance, hundreds of text words can be represented with the number of bits required to record speech with the single word "hello." This happens because text is a very economical representation of spoken words, but it excludes many other types of information. From recorded speech, on the other hand, we can identify not only the spoken words, but possibly a great deal more, such as the person's sex, age, mood, accents, and even very reliably identify the person speaking (e.g., someone can say "this is definitely my husband's voice!").

Similarly, a single X-ray image can use more data than that required to store the name, address, medical history, and other textual information of hundreds of patients. Thus, even though lossless compression of text is also important, in this book the emphasis is on compression of numerical information, since it commonly needs to be represented with a much larger number of bits.

1.3 Basic compression process

In practical applications we frequently have a mixture of data types that have to be compressed together. For example, in video compression we have to process a sequence of images together with their corresponding (synchronized) multichannel audio components. However, when starting the study of compression, it is much better to consider separately each component, in order to properly understand and exploit its particular properties. This approach is also used in practice (video standards do compress image and audio independently), and it is commonly easy to extend the basic approach and models to more complex situations.

A convenient model for beginning the study of compression is shown in Figure 1.1: we have a *data source* that generates a sequence of data elements $\{x_1, x_2, \ldots\}$, where all x_i are of the same type and each belongs to a set $\mathcal{A}$. For example, for compressing ASCII text, we can have $\mathcal{A} = \{a, b, c, \ldots, A, B, C, \ldots, 0, 1, 2, \ldots\}$. However, this representation is not always convenient, and even though this is a symbolic data type (instead of numerical), it is frequently better to use for x_i the numerical value of the byte representing the character, and have $\mathcal{A} = \{0, 1, 2, \ldots, 255\}$. For numerical data we can use as $\mathcal{A}$ intervals in the set of integer or real numbers.

The compression or *encoding* process corresponds to mapping the sequence of source symbols into the sequence $\{c_1, c_2, \ldots\}$, where each c_i belongs to a set $\mathcal{C}$ of compressed data symbols. The most common data symbols are bits ($\mathcal{C} = \{0, 1\}$) or, less frequently, bytes ($\mathcal{C} = \{0, 1, \ldots, 255\}$). The decompression or *decoding* process maps the compressed data sequence back to a sequence $\{\tilde{x}_1, \tilde{x}_2, \ldots\}$ with elements from $\mathcal{A}$. With lossless compression we always have $x_i = \tilde{x}_i$ for all i, but not necessarily with lossy compression.

There are many ways data can be organized before being encoded. Figure 1.2 shows some examples. In the case of Figure 1.2(a), groups with a fixed number of source symbols are mapped to groups with a fixed number of compressed data symbols. This approach is employed in some mathematical proofs, but is not very common. Better compression is achieved by allowing groups of different sizes. For instance, we can create a simple text compression method by using 16 bits as indexes to all text characters plus about 64,000 frequently used words (a predefined set). This corresponds to the variable-to-fixed scheme of Figure 1.2(b), where a variable number of source symbols are parsed into characters or words, and each is coded with the same number of bits. Methods for coding numerical data commonly use the method of Figure 1.2(c) where groups with a fixed number of data symbols are coded with a variable number of bits. While these fixed schemes are useful for introducing coding concepts, in practical applications it is interesting to have the maximum degree of freedom in organizing both the

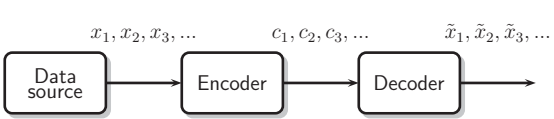

Figure 1.1 Basic data encoding and decoding model.

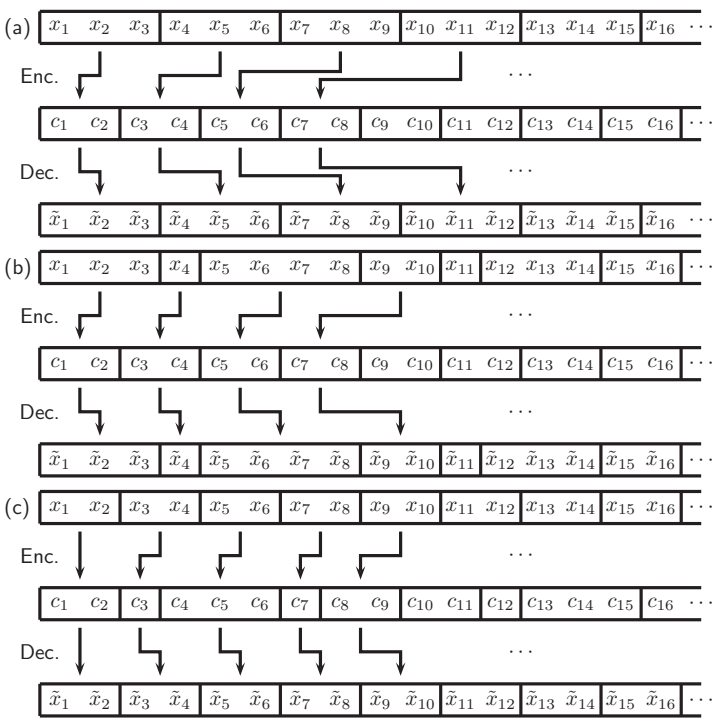

Figure 1.2 Examples of some forms in which the source and compressed data symbols can be organized for compression.

source and the compressed data symbols, so a variable-to-variable approach is more common.

1.4 Compression applications

The first popular compression applications appeared with the transition from analog to digital media formats. A great deal of research activity, and the development of the first compression standards, happened because there was a lot of analog content, such as movies, that could be converted. This is still happening, as we observe that only now is television broadcast in the USA transitioning to digital.

Another wave of applications is being created by the improvement, cost reduction, and proliferation of data-generating devices. For instance, much more digital voice traffic, photos, and even video, are being created by cell phones than with previous devices, simply because many more people carry them everywhere they go, and thus have many more opportunities to use them. Other ubiquitous personal communications and collaboration systems, such as videoconference, also have compression as a fundamental component.

The development of new sensor technologies, with the increase in resolution, precision, diversity, etc., also enables the creation of vast amounts of new digital information.

For example, in medical imaging, three-dimensional scans, which produce much more data than two-dimensional images, are becoming much more common. Similarly, better performance can be obtained by using large numbers of sensors, such as microphone or antenna arrays.

Another important trend is the explosive growth in information exchanged between devices only, instead of between devices and people. For instance, much of surveillance data is commonly stored without ever being observed by a person. In the future an even larger volume of data will be gathered to be automatically analyzed, mostly without human intervention. In fact, the deployment of sensor networks can produce previously unseen amounts of data, which may use communications resources for local processing, without necessarily being stored.

1.5 Design of compression methods

When first learning about compression, one may ask questions like:

- Why are there so many different types of compression?
- What makes a method superior in terms of performance and features?
- What compromises (if any) should be taken into account when designing a new compression method?

From the practical point of view, the subject of data compression is similar to many engineering areas, i.e., most of the basic theory had been established for many years, but research in the field is quite active because the practice is a combination of both art and science. This happens because while information theory results clearly specify optimal coding, and what are the coding performance limits, it commonly assumes the availability of reasonably accurate statistical models for the data, including model parameter values. Unfortunately, the most interesting data sources, such as text, audio, video, etc., are quite difficult to model precisely.[1]

In addition, some information theory results are asymptotic, i.e., they are obtained only when some parameter, which may be related to computational complexity or size of the data to be compressed, goes to infinity. In conclusion, while information theory is essential in providing the guidance for creating better compression algorithms, it frequently needs to be complemented with practical schemes to control computational complexity, and to create good statistical models.

It has been found that there can be great practical advantages in devising schemes that implicitly exploit our knowledge about the data being compressed. Thus, we have a large number of compression methods that were created specifically for text, executable code, speech, music, photos, tomography scans, video, etc. In all cases, assumptions were made about some typical properties of the data, i.e., the way to compress the data is in itself an implicit form of modeling the data.

Data statistical modeling is a very important stage in the design of new compression methods, since it defines what is obviously a prime design objective: reducing

the number of bits required to represent the data. However, it is certainly not the only objective. Typical additional design objectives include

- low computational complexity;
- fast search or random access to compressed data;
- reproductions scalable by quality, resolution, or bandwidth;
- error resiliency.

The need to control computational complexity is a main factor in making practical compression differ from more abstract and general methods derived from information theory. It pervades essential aspects of all the compression standards widely used, even though it is not explicitly mentioned. This also occurs throughout this book: several techniques will be presented that were motivated by the need to optimize the use of computational resources.

Depending on the type of data and compression method, it can be quite difficult to identify in compressed data some types of information that are very easily obtained in the uncompressed format. For instance, there are very efficient algorithms for searching for a given word in an uncompressed text file. These algorithms can be easily adapted if the compression method always uses the same sequence of bits to represent the letters of that word. However, advanced compression methods are not based on the simple replacement of each symbol by a fixed set of bits. We have many other techniques, such as

- data is completely rearranged for better compression;
- numerical data goes through some mathematical transformation before encoding;
- the bits assigned for each character may change depending on context and on how the encoder automatically learns about the data characteristics;
- data symbols can be represented by a fractional number of bits;
- data may be represented by pointers to other occurrences of the same (or similar) data in the data sequence itself, or in some dynamic data structures (lists, trees, stacks, etc.) created from it.

The consequence is that, in general, fast access to compressed information cannot be done in an ad hoc manner, so these capabilities can only be supported if they are planned when designing the compression method. Depending on the case, the inclusion of these data access features may degrade the compression performance or complexity, and this presents another reason for increasing the number of compression methods (or modes in a given standard).

Resiliency to errors in the compressed data can be quite important because they can have dramatic consequences, producing what is called *catastrophic error propagation*. For example, modifying a single bit in a compressed data file may cause all the subsequent bits to be decoded incorrectly. Compression methods can be designed to include techniques that facilitate error detection, and that constrains error propagation to well-defined boundaries.

1.6 Multi-disciplinary aspect

As explained above, developing and implementing compression methods involves taking many practical factors, some belonging to different disciplines, into account. In fact, one interesting aspect of practical compression is that it tends to be truly multidisciplinary. It includes concepts from coding and information theory, signal processing, computer science, and, depending on the material, specific knowledge about the data being compressed. For example, for coding image and video it is advantageous to have at least some basic knowledge about some features of the human vision, color perception, etc. The best audio coding methods exploit psychophysical properties of human hearing.

Learning about compression also covers a variety of topics, but as we show in this book, it is possible to start from basic material that is easy to understand, and progressively advance to more advanced topics, until reaching the state-of-the-art. As we show, while some topics can lead to a quite complex and advanced theory, commonly only some basic facts are needed to be used in compression.

In the next chapter we present a quick overview of the topics included in this book.

Note

1. Note that we are referring to statistical properties only, and excluding the much more complex semantic analysis.

References

1. J. F. Gantz, C. Chite, A. Manfrediz, S. Minton, D. Reinsel, W. Schliditing, and A. Torcheva, "The diverse and exploding digital universe," International Data Corporation (IDC), Framingham, MA, White paper, Mar. 2008, (http://www.emc.com/digital_universe).
2. N. Negroponte, *Being Digital*. New York, NY: Alfred A. Knopf, Inc., 1995.

2 Book overview

2.1 Entropy and lossless coding

Compression of a digital signal source is just its representation with fewer information bits than its original representation. We are excluding from compression cases when the source is trivially over-represented, such as an image with gray levels 0 to 255 written with 16 bits each when 8 bits are sufficient. The mathematical foundation of the discipline of *signal compression,* or what is more formally called *source coding,* began with the seminal paper of Claude Shannon [1, 2], entitled "A mathematical theory of communication," that established what is now called *Information Theory.* This theory sets the ultimate limits on achievable compression performance. Compression is theoretically and practically realizable even when the reconstruction of the source from the compressed representation is identical to the original. We call this kind of compression *lossless coding*. When the reconstruction is not identical to the source, we call it *lossy coding*. Shannon also introduced the discipline of *Rate-distortion Theory* [1–3], where he derived the fundamental limits in performance of lossy coding and proved that they were achievable. Lossy coding results in loss of information and hence distortion, but this distortion can be made tolerable for the given application and the loss is often necessary and unavoidable in order to satisfy transmission bandwidth and storage constraints. The payoff is that the degree of compression is often far greater than that achievable by lossless coding.

In this book, we attempt to present the principles of compression, the methods motivated by these principles, and various compression (source coding) systems that utilize these methods. We start with the theoretical foundations as laid out by Information Theory. This theory sets the framework and the language, motivates the methods of coding, provides the means to analyze these methods, and establishes ultimate bounds in their performance. Many of the theorems are presented without proof, since this book is not a primer on Information Theory, but in all cases, the consequences for compression are explained thoroughly. As befits any useful theory, the source models are simplifications of practical ones, but they point the way toward the treatment of compression of more realistic sources.

The main focus of the theoretical presentation is the definition of information entropy and its role as the smallest size in bits achievable in lossless coding of a source. The data source emits a sequence of random variables, $X_1, X_2, \ldots$, with respective probability mass functions, $q_{X_1}(x_1), q_{X_2}(x_2), \ldots$ These variables are called *letters* or *symbols* and