Chapter 11: Theory of Image Lossless & Lossy Compression

11.1 Objectives of Data Compression

Given a signal and a channel, find an encoder and decoder that give the "best possible" reconstruction. To formulate as a precise problem, we need:

- probabilistic descriptions of signal and channel (parametric model or sample training data),
- possible structural constraints on form of codes (block, sliding block, recursive),
- Quantifiable notion of what "good" or "bad" reconstruction is (PSNR, SNR, P_e *or BER*, and sometimes MOS).

Mathematical: Quantify what "best" or optimal achievable performance is.

Practical : How do you build systems that are good, if not optimal?

How to measure performance?

- SNR and Peak SNR (PSNR),
- MSE,
- P_e,
- Bit rates,
- Complexity,
- Cost
- Subjective measures, such as MOS.

Shannon Theory:

Two publications by C.E. Shannon, "A Mathematical Theory of Communication (1948)" and "Coding Theorems for a Discrete Source with a Fidelity Criterion (1957)" have provided basis for application of probability theory to solving problems in communications in addition to establishing the foundations of the Science of Modern Information Theory.

The **primary goal of Shannon's Information Theory** is to find the theoretical limits to achievable performance. That is, given a probabilistic model of an information source and a communication channel (for storage or transmission), how reliably can the source signals be communicated to a receiver though the given channel?

Note that it does not tell you how to do it?

Key to Shannon's formulation was a family of definitions of the *information* content of signals in terms of probabilities, e.g.:

Entropy:

- Assume a source S that can generate any of L possible messages.
- Let each of these messages has a probability: $P_S = \{p_k; k = 0,1,\cdots, L-1\}$
- **Entropy** of this source represent the theoretically achievable lower bound on compression Systems and it is defined by:

$$H(S)=\sum_{k=0}^{L-1} p_k.\log_2(1/p_k)=-\sum_{k=0}^{L-1} p_k.\log_2(p_k) \qquad \text{bits/symbol}$$

- Let us assume that a given source has L different messages, and
- The k^{th} symbol occurs with probability P_k, $k= 0,1,...,L-1$.
- Let us consider an encoder, which assigns a codeword with l_k bits to a particular symbol s_k.
- **Average length** of this source coder is simply:

$$L_{ave}=\sum_{k=0}^{L-1} p_k.l_k \qquad \text{bits/symbol}$$

Shannon Source Coding Theorem: Given a source S with entropy $H(S)$ then it is possible to encode this source with a distortionless source coder with an average length L_{ave} provided

$$L_{ave} \geq H(S)$$

Conversely, there is no distortionless source coder to encode S if

$$L_{ave} < H(S)$$

Channel Capacity:

- Let a source emit symbols $S_0, S_1,...,S_{L-1}$ at a rate R bits per second.
- Transmitted over a channel with a bandwidth B Hz.
- Receiver detects signals coming from the channel with a signal-to-noise ratio SNR= S/N,
- Where S and N are the signal and noise powers at the input to the receiver, respectively.
- Receiver issues symbols $Y_0,Y_1,...,Y_{K-1}$. (These symbols $\{Y_k\}$ may or may not be identical to the source set $\{S_k\}$ depending upon the nature of the receiver.) Furthermore,
- L and K may be of different size. (Some codewords might be totally lost in the channel or some codewords might be added by the channel itself.)

If the channel is **noiseless** then

- L and K are identical and
- Receiver symbols are also same as the source symbols. In this case,
- Reception of some symbol Y_k uniquely determines the source symbol S_k. (In the noisy channels, however, there is a certain amount of uncertainty regarding the identity of the transmitted symbol when Y_k is received.)
- If the information channel has a bandwidth of B Hz. and
- System is designed to operate at a signal-to-noise ratio SNR then

The highest rate we can reliably transmit binary information is called **Capacity of a noisy channel** and it is defined by:

$$C = B.\log_2(1+SNR) \quad \text{bits/symbol}$$

Shannon Channel Coding Theorem: Given a channel with a capacity C, it is possible to transmit symbols from a source emitting at a rate R bits per second with an arbitrarily small probability of error in over this channel if

$$C \geq R$$

Conversely, all systems transmitting at a rate R such that

$$C < R$$

are bound to have errors with probability one.

Implication:

We can have good source coders, even perfect ones, if the source rate is under the channel capacity. We may not be able design such good coder but that is our problem, not the information theory's! On the other hand, all coders are destined to make errors if they operate at a rate above the channel capacity.

Combination of these theorems shows that in the problem of Point-to-Point communication, the encoder (and decoder) can be decomposed into two separate pieces:

Source coding is the conversion of an information source into an efficient binary (or other digital) representation of rate R bits/s, with no regard for the channel except that its capacity is $C > R$ bits/s.

Channel coding or error control coding, takes the binary data stream of R bits/s and sends it reliably across the channel. This result is usually called the source/channel separation theorem. This result is not true in general in network communications, e.g., many sources to one receiver (multi-user channel) or one source to many receivers (broadcast channel).

- Source coding reduces the number of bits in order to save on transmission time or storage space. Compression \removes redundancy" to gain efficiency.
- Channel coding typically increases the number of bits or chooses different bits in order to protect against channel errors. Error control "adds redundancy" to permit the detection and correction of errors by looking for violation of known structures of transmitted signals and picking a likely fix.

Overall communication requires a balance of these two effects.

Fundamental Question: What does all this theory have to do with the real world?

- The theorem supports the intuitive practice that good overall systems can be designed by *separately* focusing on the source coding (compression) and channel coding (error control) problems, without concern of the interaction between the two.
- Joint coders are of increasing interest in networks and because they can yield simpler implementations.
- Theory provides benchmarks for comparison for real communication systems. It is impossible to do better than Shannon's bounds (unless you cheat). Thus if you are operating close to the Shannon limit, hardly worth doing any more.
- Most systems of the time fell far short of Shannon. However, few believed that they could ever get near. Over the years, methods evolved to come close to these limits in many applications of both source coding and channel coding.
- Emphasis through first half of Shannon era was on channel coding.
- Compression has historically played a secondary role, partially because of apparently relatively small potential gains and highly nonlinear nature in comparison with error control coding methods.

Question: Why to compress?

Theory: As systems get better, even small gains get more important.

Practice: Sometimes have no choice, must compress or lose all the data and Gains are sometimes significant.

Consider the following simple image compression tasks:

1. Low-resolution, TV quality, color video with 512*x*512 pixels/color at 8 bits/pixel, and 3 colors =~ 6 $x 10^6$ bits
2. 24 *x* 36 mm (35-mm) negative photograph scanned at 12 m which is 3000 *x* 2000 pixels/color, 8 bits/pixel, and 3 colors =~144 x 10^6 bits
3. 14 *x* 17 inch radiograph scanned at 70 m: 5000 *x* 6000 pixels, at resolution 12 bits/pixel =~ 360 x 10^6 bits. Medium size hospital generates terabytes each year.
4. Scenes from LANDSAT Thematic Mapper: 6000x6000 pixels/spectral band, 8 bits/pixel, and 6 non-thermal spectral bands =~1*:7* x 10^9 bits.

Therefore, data compression is required for **efficient transmission**

- to send more data in available bandwidth.
- to send the same data in less bandwidth
- more users on same bandwidth

and **storage**

- to store more data
- to compress for local storage, and
- to put details on cheaper media

In both cases choice may be to compress or lose, e.g., you are allotted a given storage or bandwidth by an overall systems design, and your data rate is larger than the available storage or transmission capacity.

- Compression is also useful for progressive reconstruction, scalable delivery, browsing.
- It can also speed other signal processing by reducing the number of bits to be crunched (provided we have not lost essential information) or by combining with the compression algorithm, e.g.,
    - Enhancement
    - classification/detection
    - regression/estimation
    - halftoning
    - filtering.

11.2 Lossy vs. Lossless compression

It is also known as the noiseless coding, lossless coding, invertible coding, entropy coding, and data compaction codes. They can perfectly recover original data (if no storage or transmission bit errors, i.e., noiseless channel).

- They normally have variable length binary codewords.
- Only works for *digital sources*.

General idea: Code highly probable symbols into short binary sequences, low probability symbols into long binary sequences, so that average is minimized. Most famous examples:

- **Morse code:** Consider dots and dashes as binary. Chose codeword length inversely proportional to letter relative frequencies.
- **Huffman code** 1952, employed in UNIX *compact* utility and many standards.
- **Run-length codes** popularized by Golomb in early 1960s, used in JPEG standard.
- **Lempel-Ziv(-Welch) codes** 1977,78 in Unix *compress* utility, diskdouble, stuffit, stacker, PKzip, DOS, GIF
- **Arithmetic codes** by Fano, Berlecamp, Rissanen, IBM Q-coder.

Warning: To get average rate down, need to let maximum instantaneous rate grow. This means that can get data expansion instead of compression in the short run.
Typical lossless compression ratios: 2:1 to 4:1

Lossy compression: Not invertible, information lost. However, permits more compression.
Examples:

- PCM (Shannon (1938), Oliver, Pierce 1948).
- Sampling + scalar quantization.
- Analog to digital conversion.
- By introducing loss in a controlled fashion, could prevent further loss in channel. Birth of digital communication.
- Predictive coding developed initially by Elias (1955). Code differences (residual, prediction error). Specific systems:
- Predictive scalar quantization.
  - DPCM,
  - Delta modulation,
  - Sigma Delta modulation.

  CD players use Sigma Delta demodulation, descendant of early DPCM and delta modulation, both forms of predictive codes.
- Optimal quantization Lloyd (1956) Optimizing PCM. Connection of quantization and statistics (clustering)
- **Transform coding** Mathews and Kramer (1956), Huang, Habibi, Chen (Compression Labs Inc). Dominant image coding (lossy compression) method: ITU and other standards:
  - p*64, H.261, JPEG,
  - MPEG I, II, IV, and now VII.
  - C-Cubed, CLI
  - Picture-Tel: They use transform coding (DCT) + custom uniform quantizers + runlength coding + Huffman or arithmetic coding.
  - JPEG is ubiquitous in WWW.
- **LPC** First very low bit rate speech coding based on sophisticated SP, implementable in DSP chips. (1970s) Itakura and Saito, Atal et al., Markel and (A.H.) Gray, NTT, Bell, Signal Technology, TI (Speak and Spell).
- **Subband/pyramid/wavelet coding** (late1980s). Used in MUSICAM digital audio (European standard), EZW (Knowles, Shapiro), SPHIT (Said and Pearlman), CREW (RICOH Calif.), JPEG 2000, ACR-NEMA.

Properties: (Possibly) in between lossless and lossy: "perceptually lossless" compression. (Perhaps) need not retain accuracy past:

- what the eye can see (which may depend on context),
- the noise level of the acquisition device,
- what can be squeezed through transmission or storage, i.e., an imperfect picture may be better than no picture at all.

- **Vector quantization** including code excited LPC, (CELP) (Stewart, Atal): It is imperative for HDTV systems. It is currently in use in software based video: table lookups instead of computation Apple's QuickTime, Intel's Indeo, Sun's Cell, Supermac Technology's Cinepak, Media Vision's Motive, VXtreme

  Various structures:
    - Product codes (scalar quantization, gain/shape)
    - Successive approximation/ tree-structured
    - Predictive and finite-state
    - Fractal coding, etc.

11.3 Examples

Demonstrate a number of cases using VCDEMO tool.

11.4 Is lossy compression acceptable?

NO in some applications: Computer programs, bank statements, espionage. Many have thought necessary for science and medical images. *Is it?*

Loss may be unavoidable, may have to choose between imperfect image or no data at all (or long delays). Some loss not a problem with:

- Follow-up studies
- Archives
- Research and Education
- Entertainment

Growing evidence suggests lossy compression does not damage diagnosis and may in fact improve it if it is done intelligently.

In general, lossy compression may also include a lossless compression component, to squeeze a few more bits out.

Compression application areas:

- Teleconferencing,
- FAX,
- Medical PACS (archival, educational, remote diagnosis),
- Remote sensing, Space, Remote control,
- Multimedia, Web,
- Binary, gray-scale, color, video, audio, instructional TV on the net.