GI12/4C59: Information Theory

Lectures 22–24

Massimiliano Pontil

About these lectures

Theme: We go back to the problem of source coding and discuss coding methods which are efficient when only some properties of the source distribution are known. Specifically, we prove the existence of efficient codes in the case that the entropy of the source is bounded and discuss the important Lempel-Ziv coding. To illustrate these ideas we introduce a new tool from probability called the method of types.

Outline

1. The method of types
2. Universal source coding
3. Lempel-Ziv coding

Type of a sequence

Let $X_1, \ldots, X_n$ be jointly r.v., each with values in a finite set $\mathcal{X}$ of m elements, and let x^n (also denoted by $\mathbf{x}$) be a sequence $(x_1, \ldots, x_n) \in \mathcal{X}^n$.

- The *type* $P_{\mathbf{x}}$ of a sequence $\mathbf{x}$ is the empirical distribution (relative frequency) of the symbols of $\mathcal{X}$ in $\mathbf{x}$,

$$P_{\mathbf{x}}(a) = N(a|\mathbf{x})/n, \quad a \in \mathcal{X}$$

  where $N(a|\mathbf{x})$ is the number of occurrences of symbol a in $\mathbf{x}$.
- We denote by $\mathcal{P}_n$ the set of types with denominator n.
- If $P \in \mathcal{P}_n$, we denote by $T(P)$ the set of sequences of length n and type P (or the type class of P).

Examples

Example 1: If $\mathcal{X} = \{0, 1\}$, the set of possible types of denominator n is

$$\mathcal{P}_n = \left\{ \left(\frac{0}{n}, \frac{n}{n}\right), \left(\frac{1}{n}, \frac{n-1}{n}\right), \ldots, \left(\frac{n}{n}, \frac{0}{n}\right) \right\}.$$

Example 2: Let $\mathcal{X} = \{1, 2, 3\}$. The type of $\mathbf{x} = 33311$ is $(\frac{2}{5}, 0, \frac{3}{5})$ because $P_{\mathbf{x}}(1) = \frac{2}{5}$, $P_{\mathbf{x}}(2) = 0$, $P_{\mathbf{x}}(3) = \frac{3}{5}$. The type of 11321 is $(\frac{3}{5}, \frac{1}{5}, \frac{1}{5})$.

The class of type $P = (\frac{3}{5}, \frac{1}{5}, \frac{1}{5})$ consists of the sequences of length 5 containing three 1, one 2, and one 3. There are 20 such sequences,

$$|T(P)| = \frac{5!}{3!1!1!} = 20.$$

Properties of types

(1) $|\mathcal{P}_n| \leq (n+1)^m$

(2) If $X_1, \ldots, X_n$ are i.i.d. according to $p(x)$, then

$$\text{Prob}(\mathbf{x}) = 2^{-n(H(P_{\mathbf{x}}) + D(P_{\mathbf{x}} \| p))}$$

(3) For every $P \in \mathcal{P}_n$,

$$\frac{1}{(n+1)^m} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)}$$

(4) Under the same hypothesis in (2), for every $P \in \mathcal{P}_n$

$$\frac{1}{(n+1)^m} 2^{-nD(P\|p)} \leq \text{Prob}(T(P)) \leq 2^{-nD(P\|p)}$$

Proof of 1

Remember that $m = |\mathcal{X}|$ and that $P_{\mathbf{x}}$ is identified by a vector of m components, the frequencies of symbols of $\mathcal{X}$ in $\mathbf{x}$. Since the nominator of each of these frequencies can take $n+1$ values, there are at most $(n+1)^m$ vectors.

Remark: This property says that there are a polynomial number of types of length n. Thus, since there are m^n possible sequences, there is at least one type which has exponentially many sequences.
In fact, property 3 above says that *each* type has an exponential number of sequences. The largest type class has essentially (i.e. to the first order in the exponent) the same number of sequences as the entire class.

Proof of property 2

Since $X_1, \ldots, X_n$ are i.i.d.,

$$\text{Prob}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i) = \prod_{a \in \mathcal{X}} p(a)^{N(a|\mathbf{x})}.$$

We then compute

$$\begin{aligned}
\text{Prob}(\mathbf{x}) &= \prod_{a \in \mathcal{X}} p(a)^{N(a|\mathbf{x})} \\
&= \prod_{a \in \mathcal{X}} p(a)^{nP_{\mathbf{x}}(a)} = \prod_{a \in \mathcal{X}} 2^{nP_{\mathbf{x}}(a) \log p(a)} \\
&= \prod_{a \in \mathcal{X}} 2^{nP_{\mathbf{x}}(a) \log p(a) - P_{\mathbf{x}}(a) \log P_{\mathbf{x}}(a) + P_{\mathbf{x}}(a) \log P_{\mathbf{x}}(a)} \\
&= 2^{n \sum_{a \in \mathcal{X}} (-P_{\mathbf{x}}(a) \log \frac{P_{\mathbf{x}}(a)}{p(a)} + P_{\mathbf{x}}(a) \log P_{\mathbf{x}}(a))} = 2^{n(-D(P_{\mathbf{x}}\|p) - H(P_{\mathbf{x}}))}.
\end{aligned}$$

Proof of property 3

The exact size of $T(P)$ is

$$|T(P)| = \frac{n!}{(nP(a_1))! \cdots (nP(a_m))!}.$$

The upper and lower bound can be proved by using Stirling's formula. Alternatively note, by property 2 with $p = P_{\mathbf{x}} = P$, that

$$\begin{aligned} 1 &\geq \text{Prob}(T(P)) = \sum_{\mathbf{x} \in T(P)} 2^{-nH(P)} \\ &= |T(P)| 2^{-nH(P)} \quad \Rightarrow \quad |T(P)| \leq 2^{nH(P)} \end{aligned}$$

The lower bound requires longer (but similar to the above) computations.

Proof of property 4

Using property 3, we have

$$\begin{aligned} \text{Prob}(T(P)) &= \sum_{\mathbf{x} \in T(P)} \text{Prob}(\mathbf{x}) = \sum_{\mathbf{x} \in T(P)} 2^{-n(H(P)+D(P\|p))} \\ &= |T(P)| 2^{-n(H(P)+D(P\|p))} \end{aligned}$$

and, by property 3, we conclude that

$$\frac{1}{(n+1)^m} 2^{-nD(P\|p)} \leq \text{Prob}(T(P)) \leq 2^{-nD(P\|p)}.$$

Types: wrapping up

We have seen that

- there is only a polynomial (in n) number of types (property 1) and each type class has exponentially many elements (property 3).
- For i.i.d. random variables, property 4 says that type classes P that are far from the true distribution p have exponentially smaller probability.

This observation motivates the following definition: given $\epsilon > 0$ we define, for n large enough, the *typical set*

$$T_\epsilon^{(n)} = \{x^n : D(P_{x^n} \parallel p) \leq \epsilon\}.$$

An useful lemma

Lemma: If $X_1, \ldots, X_n$ are i.i.d. $\sim p(x)$, then

$$\text{Prob}(\{D(P_{x^n} \parallel p) > \epsilon\}) \leq \frac{1}{(n+1)^m} 2^{-n\epsilon} = 2^{-n\epsilon - m\log(n+1)}.$$

$$\begin{aligned}
\text{Prob}(\{D(P_{x^n} \parallel p) > \epsilon\}) &= \sum_{P: D(P\|p) > \epsilon} \text{Prob}(T(P)) \\
&\leq \sum_{P: D(P\|p) > \epsilon} 2^{-nD(P\|p)} \qquad \text{(by property 4)} \\
&\leq \sum_{P: D(P\|p) > \epsilon} 2^{-n\epsilon} \\
&\leq \frac{1}{(n+1)^m} 2^{-n\epsilon} \qquad \text{(by property 1).}
\end{aligned}$$

The typical set

Lemma: If $X_1, \ldots, X_n$ are i.i.d. $\sim p(x)$, then

$$\text{Prob}(\{D(P_{x^n} \parallel p) > \epsilon\}) \leq 2^{-n\epsilon - m\log(n+1)}.$$

The lemma implies that the probability of a typical set

$$T_\epsilon^{(n)} = \{x^n : D(P_{x^n} \parallel p) \leq \epsilon\}$$

converges to 1 as $n \to \infty$.

We also have

$$\sum_{n=1}^{\infty} P(\{D(P_{x^n} \parallel p) > \epsilon\}) < \infty$$

That is, the expected number of occurrences of the event $\{D(P_{x^n} \parallel p) > \epsilon\}$ for all n is finite. This implies (skip proof) that

$$D(P_{x^n} \parallel p) \to 0 \quad \text{with probability 1.}$$

Two notions of convergence

Remember that the sequence of r.v. $\{T_n : n \in \mathbb{N}\}$ converges

- *in probability* to a r.v T if, for every $t > 0$, we have that

$$\lim_{n\to\infty} P(\{|T_n - T| \geq t\}) = 0$$

- *with probability* 1 to a r.v. T if the following holds

$$P\left(\left\{\omega \in \Omega : \lim_{n\to\infty} T_n(\omega) = T(\omega)\right\}\right) = 1$$

Convergence in probability (aka almost sure convergence) implies convergence in probability.

Universal source coding

We wish to compress an i.i.d. source when the distribution p of the source is unknown but has bounded entropy, say $H(p) < R$. This scenario is common in practice...

If we use, say, an Huffman code to compress the source sequences and this code is designed for a distribution $q(x)$, we may pay a penalty of $D(p^{(n)} \parallel q^{(n)})$, where $p^{(n)}(x^n) = p(x_1) \cdots p(x_n)$. That is,

$$E_p[\ell(X_1, \ldots, X_n)] - E_q[\ell(X_1, \ldots, X_n)] \approx D(p^{(n)} \parallel q^{(n)}) = nD(p \parallel q).$$

- *can we do better?*

Universal source coding (cont)

The goal is to design a code which is "efficient" for every p such that $H(p) < R$.

A *fixed rate block code* of rate R consists of an encoder $f_n : \mathcal{X}^n \to \{1, 2, \ldots, 2^{nR}\}$, and a decoder $g_n : \{1, 2, \ldots, 2^{nR}\} \to \mathcal{X}^n$. (These two mappings are independent of p.) The probability of error of the code is

$$P^{(n)} = \text{Prob}(g_n(f_n(X_1, \ldots, X_n)) \neq (X_1, \ldots, X_n)).$$

Such code is called *universal* if $P^{(n)} \to 0$ as $n \to \infty$ for every p such that $H(p) < R$.

Remark: this code is lossy (singular), since $R < \log m$ and, so, $2^{nR} < m^n$.

Universal codes exist

Idea: we set $R_n = R - m\frac{\log(n+1)}{n}$ and consider the set of sequences

$$A = \{\mathbf{x} \in \mathcal{X}^n : H(P_{\mathbf{x}}) < R_n)\}.$$

We show below using the method of types that Prob$(A) \to 1$ as $n \to \infty$ and $|A| \leq 2^{nR}$.

Informal justification: If n is large enough

- there are about $2^{nH(P)}$ sequences of type P and, since there are only a polynomial number of types with denominator n, we need roughly nR bits to enumerate all sequences such that $H(P_{\mathbf{x}}) < R$.
- the remaining sequences have small probability.

Choice of the functions f_n and g_n

We index the elements of A and define the encoder f_n as

$$f_n(\mathbf{x}) = \begin{cases} index(\mathbf{x}) & \text{if } \mathbf{x} \in A \\ 0 & \text{otherwise .} \end{cases}$$

We also define the decoder g_n to map each index to the corresponding element of A or to a $dummy$ element otherwise.

Thus, the probability of error is

$$P^{(n)} = \text{Prob}((X_1, \ldots, X_n) \notin A).$$

Proof of the technical step 1

We now prove the above technical steps.

In order to prove that $|A| \leq 2^{nR}$ we set

$$\mathcal{P}' = \left\{ P \in \mathcal{P}_n : H(P) \leq R_n = R - m\frac{\log(n+1)}{n} \right\}$$

and note, by property 1 and 2 above, that

$$\begin{aligned} |A| &= \sum_{P \in \mathcal{P}'} |T(P)| \leq \sum_{P \in \mathcal{P}'} 2^{nH(P)} \leq \sum_{P \in \mathcal{P}'} 2^{nR_n} \\ &\leq (n+1)^m 2^{nR_n} = 2^{n(R_n + \frac{m}{n}\log(n+1))} = 2^{nR}. \end{aligned}$$

Proof of the technical step 2

We now show that $P^{(n)} \to 0$. Using properties 1 and 2, we derive

$$\begin{aligned} P^{(n)} &= \sum_{P \in \bar{\mathcal{P}}'} \text{Prob}(T(P)) \leq (n+1)^m \max_P \left\{ \text{Prob}(T(P)) : \ P \in \bar{\mathcal{P}}' \right\} \\ &\leq (n+1)^m 2^{-n \min_P \{ D(P\|p) : \ P \in \bar{\mathcal{P}}' \}}. \end{aligned}$$

Then, we note that, since $R_n \uparrow R$ and $H(p) < R$, for n large enough, we have $R_n > H(p)$ and, so,

$$\min_P \left\{ D(P \parallel p) : \ P \in \bar{\mathcal{P}}' \right\} = \min_P \{ D(P \parallel p) : \ H(P) > R_n \} > 0$$

which implies that $P^{(n)} \to 0$ as $n \to \infty$.

Some remarks

- We pay an extra price in designing an universal code: to obtain the same performance as, say, an Huffman code, we need a longer block length.

- The above proof technique also tells us that if $H(p) > R$, the sequence will have a type outside A and the probability of error will be close to 1.

- Proof technique is restricted to i.i.d. sources.

Lempel-Ziv Coding

We now remove the i.i.d. assumption and discuss an universal code, named Lempel-Ziv code, for more general sources.

We discuss this code for binary alphabets but its extension to finite alphabets is immediate.

Consider the following example

Example: Let $\mathbf{x} = 1011010100010\ldots$ We parse $\mathbf{x}$ as a sequence of successive substrings of it

$$1, 0, 11, 01, 010, 00, 10$$

What's the idea behind this?

Extracting the parse sequence

In the above example, we have sequentially extracted strings which "have not appeared so far". More precisely, the parsed sequence is obtained by extracting, after each comma, an input subsequence which

(1) starts at the end of the previous subsequence, and

(2) is the shortest string which has not been extracted before.

Remark: A consequence of (2) is that all prefixes of an extracted string were extracted before this string (e.g., the prefixes of 010 are 0 and 01 and were both extracted before 010).

Coding

Let n be the length of $\mathbf{x}$ and $c(n)$ the number of strings in the parse sequence $y_1, y_2, \ldots, y_{c(n)}$. The code of $\mathbf{x}$ is identifies by

- a pointer codeword of $\log c(n)$ bits which codes for the prefix of y_j, $j = 1, \ldots, c(n)$.
- a single bit which describes the last bit in y_j.

Example: Let $\mathbf{x} = 1011010100010$. The parsed sequence is $1, 0, 11, 01, 010, 00, 10$ and the code of $\mathbf{x}$ is

$$(000, 1), (000, 0), (001, 1), (010, 1), (100, 0), (010, 0), (001, 0)$$

Decoding

Given a Lempel-Ziv codeword, we decode it into a string $x^n = (y_1, \ldots, y_c)$, by computing the y_i sequentially.

In particular, $(000,1),(000,0),(001,1),(010,1),(100,0),(010,0),(001,0)$ is decoded as

Iteration	(p-code,l-bit)	Parse-string	Sequence
0	$(\emptyset,\emptyset)$	$y_0 = \emptyset$	$\emptyset$
1	(000,1)	$y_1 = y_0 1 = 1$	**1**
2	(000,0)	$y_2 = y_0 0 = 0$	1**0**
3	(001,1)	$y_3 = y_1 1 = 11$	10**11**
4	(010,1)	$y_4 = y_2 1 = 01$	1011**01**
5	(100,0)	$y_5 = y_4 0 = 010$	101101**010**
6	(010,0)	$y_6 = y_2 0 = 00$	10110101**00**
7	(001,0)	$y_7 = y_1 0 = 10$	1011010100**10**

Some observations

- The Lempel-Ziv code is a variable length block code. The function f_n (which was now specified before) maps $\mathcal{X}^n$ into $\mathcal{X}^*$.

- By construction the code is non-singular and, so, $P^{(n)} = 0$ for every n.

- In the above example the Lempel-Ziv code is about twice longer than $\mathbf{x}$. However, for longer strings the code will compress $\mathbf{x}$. In fact, as we will show below, this code is asymptotically optimal. This also implies that the code is universal.

Some observations (cont.)

- To compute the code we need two passes through the input string $\mathbf{x}$. In the first pass we produce the parsed sequence and calculate $c(n)$. In the second pass, we calculate the pointers.

- It is possible to modify the Lempel-Ziv code to: i) have variable length pointers (in fact, the first pointers need fewer bits) and ii) have only a single pass through $\mathbf{x}$. These changes do not affect the asymptotic properties of the code.

Optimality of Lempel-Ziv coding

The total length of the compress sequence is

$$\ell(x^n) = c(n)(\log c(n) + 1).$$

We wish to show that the code is *universal* and *asymptotically optimal*, namely

$$\frac{\ell(X^n)}{n} \to H(\mathcal{X}) \quad \text{with probability 1}$$

where $H(\mathcal{X})$ is the entropy rate of the process.

Plan for the proof

The proof is based on the following main steps

- Find an upper bound for $c(n)$.
- Approximate the process by a $k-$order Markov process.
- Bound the entropy for such a process (Lemma B and C), thereby proving optimality.
- Observe that the approximation is exact in the limit $k \to \infty$.

Upper bound on $c(n)$

Formally, a parse of a string x^n is a division of x^n into contiguous subsequences of it (phrases) separated by a comma. We assume here that these subsequences are always distinct. For example, the Lempel-Ziv parser gives distinct subsequences.

Lemma A: If the parser is distinct, we have

$$c(n) \leq \frac{n}{(1-\epsilon_n)\log n}$$

where $\epsilon_n > 0$ and $\epsilon_n \to 0$ as $n \to \infty$.

Remark: Note that $c(n)$ is function of the sequence x^n and the above results holds for *every* sequence, that is

$$\sup_{x^n} c(n) \leq \frac{n}{(1-\epsilon_n)\log n}.$$

For a proof of the lemma see pp. 321 of Cover and Thomas's.

$k-$order Markov process

We prove the optimality of the code for a $k-$order Markov process. Let x_i^j denote the sequence $(x_i, \ldots, x_j)$ with $i \leq j$. We define, for $k \geq 1$,

$$Q_k(x_{1-k}, \ldots, x_0, x_1, \ldots, x_n) = P(x_{1-k}^0) \prod_{j=1}^{n} P(x_j | x_{j-k}^{j-1})$$

The initial state x_{1-k}^0 is part of specification of the process. The entropy rate of the process is

$$H(X_k | X_0^{k-1}) = \lim_{n \to \infty} -\frac{1}{n} \log Q_k(X_1, \ldots, X_n | X_{1-k}, \ldots, X_0)$$

Remark: This method of proof is general since any ergodic process can be approximate by a $k-$order Markov process. In particular, $H(X_k | X_0^{k-1}) \to H(\mathcal{X})$ for $k \to \infty$.

$k-$order Markov process (cont.)

Let x_1^n be a sampled sequence and y_1^c its parsing,

$$y_i = x_{\nu_i}^{\nu_{i+1}-1}$$

where, for $i = 1, \ldots, c$, ν_i is the index where phrase y_i starts and $\nu_{c+1} = n + 1$. We define, for $i = 1, \ldots, c$, the "preceding state" variable

$$s_i = x_{\nu_i - k}^{\nu_i - 1} \qquad \text{(the } k \text{ bits preceding } y_i\text{)}$$

and $c_{\ell s}$ be the number of phrases y_i with length ℓ and preceding state s. Then, by construction we have

$$\sum_{\ell, s} c_{\ell s} = c, \qquad \text{and} \quad \sum_{\ell, s} \ell c_{\ell s} = n \qquad (*)$$

$k-$order Markov process (cont.)

Lemma B (Ziv's inequality): $\log Q_k(x_1^n|s_1) \leq -\sum_{\ell,s} c_{\ell s} \log c_{\ell s}$.

Proof: Using the chain rule and the $k-order$ Markov property, we write

$$Q_k(x_1^n|s_1) = Q_k(y_1^c|s_1) = \prod_{i=1}^{c} P(y_i|s_i, y_1^{i-1}) = \prod_{i=1}^{c} P(y_i|s_i).$$

We define the index set $I(s,\ell) = \{i : |y_i| = \ell,\ s_i = s\}$ and note that

$$\begin{aligned} \log Q_k(x_1^n|s_1) &= \sum_{i=1}^{c} \log P(y_i|s_i) = \sum_{\ell,s} \sum_{i \in I(s,\ell)} \log P(y_i|s_i) \\ &= \sum_{\ell,s} c_{\ell s} \sum_{i \in I(s,\ell)} \log \frac{1}{c_{\ell s}} P(y_i|s_i) \leq \sum_{\ell,s} c_{\ell s} \log \left(\sum_{i \in I(s,\ell)} \frac{1}{c_{\ell s}} P(y_i|s_i) \right). \end{aligned}$$

where we used Jensen's inequality. Now the result follows by noting that, since the y_i are distinct, the inner sum is bounded by $\log 1/c_{\ell s}$.

Optimality of Lempel-Ziv code

We define $\pi_{\ell s} = c_{\ell s}/c$ and note that, by eqs. $(*)$

$$\sum_{\ell,s} \sum_{\ell,s} \pi_{\ell s} = 1 \quad \sum_{\ell,s} \ell \pi_{\ell s} = \frac{n}{c}.$$

Thus, we have

$$\log Q_k(x_1^n|s_1) \leq -\sum_{\ell,s} c_{\ell s} \log c_{\ell s} = -c \log c - c \sum_{\ell,s} \pi_{\ell s} \log \pi_{\ell s} = -c \log c + cH(U,V) \quad (**)$$

where U, V are r.v. with $\text{Prob}(U = \ell, V = s) = \pi_{\ell s}$.

Now, $H(U,V) \leq H(U) + H(V) \leq H(U) + k \log m$.

Bounding $H(U)$

We used following result to bound $H(U)$.

Lemma C: If Z is an integer and positive r.v. with mean μ, its entropy is bounded by $H(Z) \leq (\mu+1)\log(\mu+1) - \mu \log \mu$

Then, since $E(U) = \sum_{\ell,s} \ell \pi_{\ell s} = n/c$, we have

$$H(U) \leq \left(\frac{n}{c}+1\right)\log\left(\frac{n}{c}+1\right) - \frac{n}{c}\log\frac{n}{c} \pm \log\frac{n}{c} = \log\frac{n}{c} + \left(\frac{n}{c}+1\right)\log\left(\frac{c}{n}+1\right)$$

and, so,

$$\frac{c}{n}H(U,V) \leq \frac{c}{n}k\log m + \frac{c}{n}\log\frac{n}{c} + O(1) \qquad (***)$$

Concluding the proof

We rewrite inequality $(**)$ as $c\dfrac{\log c}{n} \leq -\dfrac{1}{n}\log Q_k(x_1^n|s_1) + \dfrac{c}{n}H(U,V)$.

We then observe that

- $\frac{c}{n}H(U,V) \to 0$ as $n \to \infty$ (by inequality (***) and Lemma A)
- $\lim_{n\to\infty} \sup c(n)/n = 0$ (by Lemma A)
- $\lim_{n\to\infty} \frac{1}{n}\log Q_k(x_1^n|s_1) = H(X_0|X_{-1},\ldots,X_{1-k})$.

Thus, with probability 1, we have

$$\begin{aligned}
\limsup_{n\to\infty} \frac{\ell(x^n)}{n} &\leq \limsup_{n\to\infty}\left(c(n)\frac{\log c(n)}{n} + \frac{c(n)}{n}\right) \\
&\leq \limsup_{n\to\infty} c(n)\frac{\log c(n)}{n} + \limsup_{n\to\infty}\frac{c(n)}{n} \leq \limsup_{n\to\infty} c(n)\frac{\log c(n)}{n} \\
&\leq H(X_0|X_{-1},\ldots,X_{1-k}) \quad\to\quad H(\mathcal{X}) \quad \text{as} \quad k\to\infty.
\end{aligned}$$

Bibliography

This lectures are based on Sections 12.1-3 and 12.10 of Cover and Thomas's book.