10-704: Information Processing and Learning **Spring 2012**

Lecture 6: Source coding, AEP and Typicality

Lecturer: Aarti Singh *Scribes: Mark McCartney*

6.1 Data compression/source coding

i.i.d. source p(x) $\overset{x_1,\ldots,x_n}{\rightarrow}$ compressor $\overset{nH(x)}{\rightarrow}$

Source code: A source code C is a mapping from the range of a random variable or a set of random variables to finite length strings of symbols from a D-ary alphabet.

Expected length of a source code denoted by $L(C)$ is given as follows:

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$$

where $l(x)$ is the length of codeword $c(x)$ for a symbol $x \in \mathcal{X}$, and $p(x)$ is the probability of the symbol.

Intuitively, a good code should preserve the information content of an outcome. Since information content depends on the probability of the outcome (it is higher if probability is lower, or equivalently if the outcome is very uncertain), a good codeword will use fewer bits to encode a certain or high probability outcome and more bits to encode a low probability outcome. Thus, we expect that the smallest expected code length should be related to the average uncertainty of the random variable i.e. the entropy.

We will show that entropy is the fundamental limit of data compression; i.e., $\forall C : L(C) \geq H(X)$ (source coding theorem)

Instead of encoding individual symbols, we can also encode blocks of symbols together. A length n **block code** encodes n length strings of symbols together and is denotes by $C(x_1, \ldots, x_n) =: C(x^n)$.

First, we will show that there exists a length n block code (an impractical one) with expected code length for a symbol that is arbitrarily close to entropy as $n \to \infty$. This argument is due to Shannon as presented in his seminal paper (available at http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html).

But first we need to introduce some new concepts.

6.2 Asymptotic equipartition property (AEP) & Typical Sets

Theorem 6.1 *(AEP) If* $X_1, \ldots, X_n \overset{\text{iid}}{\sim} p(x)$*, then*

$$-\frac{1}{n} \log p(x_1, \ldots, x_n) \to H(x) \text{ in probability}$$

AEP is essentially the information theoretic version of the Law of Large Numbers. The proof follows directly from the weak law of large numbers.

We will categorize n-length sequence of symbols into two types: **typical** and **atypical**. First, we discuss an intuitive notion of typical sequence. If the probabilities of symbols are given as $p_1, p_2, \ldots$ and sequences are drawn i.i.d., then we expect symbol 1 to occur $n \times p_1$ times in a typical sequence x^n. Thus, for a typical sequence

$$p(x^{(n)}) = p(x_1) \ldots p(x_n)$$
$$p(x^{(n)}) = {p_1}^{p_1 n} {p_2}^{p_2 n} {p_3}^{p_3 n} \ldots {p_{|\mathcal{X}|}}^{p_{|\mathcal{X}|} n}$$

Thus, the Shannon information content of a typical sequence:

$$\begin{aligned} -\log_2 \frac{1}{p(x^n)} &= -\log_2 \frac{1}{p_1^{p_1 n} p_2^{p_2 n} \ldots} \\ &= -n \sum_i p_i \log \frac{1}{p_i} \\ &= nH(x) \end{aligned}$$

Another way of expressing this:

$$p(x^{(n)}) = 2^{-nH(x)}$$

We now formally state the definition of a typical sequence. A **Typical set** denoted ${A_\varepsilon}^{(n)}$ w.r.t. p(x) is the set of sequences $x^n \in \mathcal{X}^n$ that satisfy the following:

$$2^{-n(H(x)+\varepsilon)} \leq p(x_1, \ldots, x_n) \leq 2^{-n(H(x)-\varepsilon))}$$

Or, equivalently:

$$A_\varepsilon^{(n)} = \left\{ x_n \in X^n : p(x^n) \in [2^{-n(H(x)+\varepsilon)}, 2^{-n(H(x)-\varepsilon)}] \right\}$$

Properties of typical set:

1. If $(x_1, \ldots, x_n) \in A_\varepsilon^{(n)}$ then $H(X) - \varepsilon \leq -\frac{1}{n} \log p\,(x_1, \ldots, x_n) \leq H(X) + \varepsilon$.
2. $Pr\left(A_\varepsilon^{(n)}\right) > 1 - \varepsilon$ for n sufficiently large.
3. $|A_\varepsilon^{(n)}| \leq 2^{n(H(x)+\varepsilon)}$
4. $|A_\varepsilon^{(n)}| \geq (1-\varepsilon) 2^{n(H(x)-\varepsilon)}$ for n sufficiently large.

Proof of property 1: Follows from definition.

Proof of property 2: By AEP, convergence in probability $\Rightarrow$ for any $\epsilon > 0$, there exists n_0 s.t. $\forall n \geq n_0$, $Pr(A_\varepsilon^{(n)}) > 1 - \epsilon$.

Proof of property 3:

$$1 = \sum_{x^n} p(x^n) \geq \sum_{x^n \in A_\varepsilon^{(n)}} p(x^n) \geq 2^{-n(H(x)+\varepsilon)} |A_\varepsilon^{(n)}|$$

Proof of property 4: By property 2 for sufficiently large n, we have

$$1 - \varepsilon < Pr(A_\varepsilon^{(n)}) = \sum_{x^n \in A_\varepsilon^{(n)}} p\,(x^n) \leq 2^{-n(H(x)-\varepsilon)} |A_\varepsilon^{(n)}|$$

6.3 Using AEP for Data Compression

Given length n i.i.d. sequences drawn from $p(x)$, we encode a sequence x^n as follows. We use one bit to indicate whether or not the sequence is typical, and then use either the index of the sequence in the typical set (if it is typical) or the index of the sequence in the entire set of length n sequences (if the sequence is atypical).

If $x^n \in A_\varepsilon^{(n)}$, we encode the sequence with the bit 0 and the index into the typical set. This requires at most $1 + \lceil n\,(H+\varepsilon)\rceil$ bits. The last term is $\log_2$ of the size of the typical set plus one bit if $\log_2 |A_\varepsilon^{(n)}|$ is not an integer.
If $x^n \notin A_\varepsilon^{(n)}$, we encode the sequence with the bit 1 and the index of the sequence in the set of all length n sequences. This requires $1 + \lceil n \log |\mathcal{X}|\rceil$ bits since the set of all length n sequences is of size $|\mathcal{X}|^n$. This can be improved, but suffices to show the result.

Notice that the above code is one-to-one and uniquely decodable. Also notice that typical sequences are encoded using much fewer bits than atypical ones if the entropy of the source H is small. (Recall that entropy is maximum $\log_2 |\mathcal{X}|$ if the source distribution is uniform.

Expected length of codeword:

$$\begin{aligned} E[l(X^n)] = \sum_{x^n} p(x^n) l(x^n) &\leq \sum_{x^n \in A_\varepsilon^{(n)}} p(x^n)\,(2 + n(H+\varepsilon)) + \sum_{x^n \notin A_\varepsilon^{(n)}} p(x^n)\,(2 + n\log|\mathcal{X}|) \\ &\leq 2 + n(H+\varepsilon) + n\log|\mathcal{X}|(1 - Pr(A_\varepsilon^{(n)})) \\ &\leq 2 + n(H+\varepsilon) + n\log|\mathcal{X}|\varepsilon \quad \text{(for } n \text{ sufficiently large)} \\ &=: n(H+\varepsilon') \\ &\quad \text{where } \varepsilon' = \varepsilon + \varepsilon\log|X| + \frac{2}{n} \end{aligned}$$

Thus, the code we constructed has expected length per symbol arbitrarily close to entropy for large n.

Theorem 6.2 *Let $x_1, \ldots, x_n \overset{\text{iid}}{\sim} p(x)$. Let $\varepsilon > 0$. $\exists$ a code which maps sequences of length n into binary strings such that mapping is one-to-one and*

$$E\left[\frac{l(x^n)}{n}\right] \leq H(x) + \varepsilon'$$

for n sufficiently large.

Thus, it is possible to compress a length n sequence $x^n = (x_1, \ldots, x_n)$ using $nH(x)$ bits on average.

While this establishes that the fundamental limit of data compression (entropy) is achieveable for i.i.d. sources, the code we have used is impractical since it requires maintaining lookup tables at the encoder and decoder that are expoentially large (i.e. size is $\approx 2^{nH(X)}$).

6.3.1 High probability sets and typical sets

We might wonder if the typical set was special, or can we come up with another high probability set with smaller size, so that we can compress sequences in that high probability set using even fewer number of bits.

Let $B_\epsilon^{(n)} \subset \mathcal{X}^n$ be the smallest set such that $Pr\left(B_\epsilon^{(n)}\right) \geq 1 - \epsilon$. This is the ideal set we would like to use for encoding, however we use the typical set since we can bound the size of the typical set more easily.

And in fact the following theorem shows that using the typical set isn't supoptimal since all other sets with probability $> 1 - \epsilon$ have about the same size.

Remark: Notice that there are sequences which belong to $B_\epsilon^{(n)}$ but are not typical, e.g. consider a source Bernoulli(0.9). Then the sequence with all 1s is the most likely (highest probability) sequence and is in $B_\epsilon^{(n)}$, however it is not a typical sequence since a typical sequence has about 0.9 as the proportion of 1s. However, such sequences contribute negligible probability.

Theorem 6.3 *Let* $x_1, \ldots, x_n \overset{\text{iid}}{\sim} p(x)$. *Let* $C_\delta^{(n)} \subset \mathcal{X}^n$. *Let* $\delta < \frac{1}{2}$ *and* $\delta' > 0$.

If $Pr\left(C_\delta^{(n)} > 1 - \delta\right)$, *then* $\frac{1}{n} \log |C_\delta^{(n)}| > H - \delta'$ *for* n *sufficiently large.*

The theorem implies that the typical set size is of the same order as any high probability set: $|A_\varepsilon^{(n)}| \approx |C_\delta^{(n)}| \approx 2^{nH}$. More specifically, $\lim_{n\to\infty} \frac{1}{n} \log \frac{C_\delta^{(n)}}{A_\delta^{(n)}} = 0$.

Proof: Let $\varepsilon < \frac{1}{2}$.

1. First, we argue that the set $C_\delta^{(n)}$ has high overlap with the typical set.

$$\begin{aligned} Pr(A_\varepsilon^{(n)} \cap C_\delta^{(n)}) &= 1 - P({A_\varepsilon^{(n)}}^C \cup {C_\delta^{(n)}}^C) \\ &\geq 1 - P({A_\varepsilon^{(n)}}^C) - P({C_\delta^{(n)}}^C) \\ &\geq 1 - \varepsilon - \delta \end{aligned}$$

2. Now we lower bound the size of $C_\delta^{(n)}$.

$$Pr(A_\varepsilon^{(n)} \cap C_\delta^{(n)}) = \sum_{x^n \in A_\varepsilon^{(n)} \cap C_\delta^{(n)}} p(x^n) \leq 2^{-n(H-\varepsilon)} |C_\delta^{(n)}|.$$

Combining 1 and 2, we have:

$$|C_\delta^{(n)}| \geq (1 - \varepsilon - \delta)\, 2^{n(H-\varepsilon)}$$

or equivalently,

$$\frac{1}{n} \log |C_\delta^{(n)}| \geq H - \varepsilon + \frac{1}{n} \log(1 - \varepsilon - \delta) \geq H - \delta'$$

for n sufficiently large.

6.4 Symbol coding

Now we'll discuss some more practical coding schemes. First, lets note that a lossless coding scheme needs to satisfy the following requirement:

Non-singular code: $x_1 \neq x_2 \implies C(x_1) \neq C(x_2)$

Since we usually have to send a stream of symbols and not a single symbol, we need the following:

Extension of a code: $C(x_1, \ldots, x_n) = C(x_1) \ldots C(x_n)$
Unique decodability: Extension is non-singular i.e. $x_i^n \neq x_j^m \implies C(x_i^n) \neq C(x_j^m)$

Moreover, Unique decodability is not enough if you have to wait until the end of the message to decode. For example, assume $C(X)$ uses the code shown in Table 6.1.

x	1	2	3	4
C(x)	10	00	11	110

Table 6.1: A code which might change early symbols after decoding later symbols

The message 110000 decodes to 322, while the message 1100000 decodes to 422, changing the first symbol two symbols after it has been received. The class of codes which does not have this problem is known as prefix, instantaneous, or self-puncturing codes. For an example of a prefix code, see Table 6.2.

x	1	2	3	4
C(x)	0	10	110	111

Table 6.2: An example of a prefix code