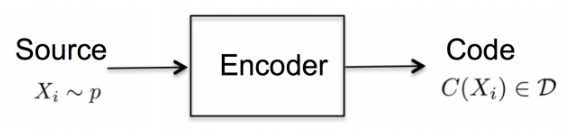
**10-704: Information Processing and Learning** **Spring 2015**

# Lecture 9: February 10

*Lecturer: Akshay Krishnamurthy* *Scribes: Rohan Varma*

**Note**: *LaTeX template courtesy of UC Berkeley EECS dept.*

## 9.1 Source Coding



A source code for a random variable $X$ is a mapping from $\mathcal{X}$, (the range of the random variable $X$) to $D^*$ , the set of all finite length strings belonging to an alphabet with D letters. We denote the codeword for $x$ as $C(x)$ and $l(x)$ as the length of the codeword for $x$.

We typically assume that $p$ , the probability distribution over $X$ is known. Hence, we aim to build a code for a given $p$ with a short length in expectation:

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

. For the following , we assume that $D = \{0, 1\}$is a binary alphabet.

### 9.1.1 Characterization of Variable-Length Codes

- Non Singular Codes: $x_1 \neq x_2 \Rightarrow C(x_1) \neq C(x_2)$ . This is an necessary property for a lossless and perfectly decodable code. We note that we can easily extend a code to $n$ input symbols.
- Extension: $C : \mathcal{X}^* \rightarrow \mathcal{D}^*$ As a result, $C(x_1, x_2, x_3, \cdots x_n) = C(x_1)C(x_2) \cdots C(x_n)$
- Unique Decodability: A code is uniquely decodable if the extension is non-singular. i.e $C(x_i^n) \neq C(x_j^m)$ for sequences $x_i^n \neq x_i^m$.
- Prefix Code: (self-punctuation or instantaneous) . $\forall x_i, x_j \in \mathcal{X}$ , $x_i \neq x_j$, $C(x_i) \neq Prefix(C(x_j))$. This implies that we can decode a sequence on the fly

### 9.1.2 Examples of Codes

We shall first informally state the Source Coding Theorem that we work towards in this lecture. Asymptotically $H(X) \leq L(C)$. $\forall$ codes $C$ $\exists$ a code (with $L(C) \leq H(X)$ ). We shall precisely state and prove this idea. Since this is an asymptotic result we need to code a sequence.

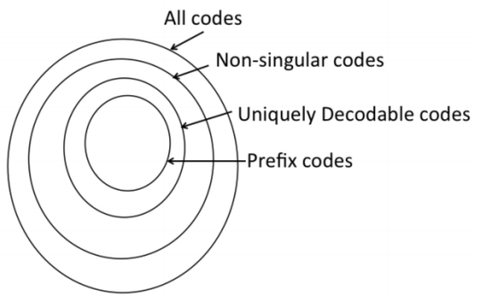


Figure 9.1: Types of Variable-Length Codes

- Encode a uniform distribution on 4 letters: We simply assign 2 bits per symbol. Hence, $\mathbb{E}l(x) = 2 = H(x)$.
- Uniform distribution on 7 letters: We assign 3 bits per letter. Hence $\mathbb{E}l(x) = 3 \leq H(x) + 1$
- $p(a) = \frac{1}{2}, p(b) = \frac{1}{4}, p(c) = \frac{1}{4}$. We code $a \to 0$, $b \to 10$, $c \to 11$. We have $\mathbb{E}l(x) = \frac{3}{2} = H(x)$. We note that is a prefix free code.
- Non-Singular Code: $p(a) = 0.5, p(b) = 0.25, p(c) = p(d) = \frac{1}{8}$, We code $a \to 0$, $b \to 1$, $c \to 00$, $d \to 01$. This is nonsingular but not very useful or practical since it is not uniquely decodable since the string $'01'$ can map to both $ab$ or $c$ . $\mathbb{E}l(x) = 1.25 \leq H(X) = 1.75$
- We code $a \to 10$, $b \to 00$, $c \to 11$, $d \to 110$. This is nonsingular and uniquely decodable but not prefix free since the code for $c$, $'11'$ is a prefix for the code for $d,' 110'$. This would result in decoding "on the fly" impossible.
- We code $a \to 0$, $b \to 10$, $c \to 110$, $d \to 111$. Hence this code is prefix-free, and therefore non-singular and uniquely decodable.

It is clear that the bare minimum requirement when designing a practical code is that it needs to be uniquely decodable.

We note that $H(X)$ seems to be the achievable rate, but because of rounding effects over the number of bits, it might not be achievable. As a result we seek to amortize over long blocks of symbols to get closer to the entropy rate.

## 9.2 Shannon Source Coding Theorem (Achievability)

We now state and prove the Shannon source coding theorem:

**Theorem 9.1** *$\exists$ a code $C$ such that for any $\epsilon \geq 0$, $\exists n_0$ such that $\forall n \geq n_0$ we have :*

$$\mathbb{E}[\frac{l(x^n)}{n}] \leq H(x) + \epsilon'$$

**Proof:** Suppose I could find a set $A_\delta^{(n)} \subseteq \mathcal{X}^{(n)}$ with $|A_\delta^{(n)}| \leq 2^{n(H(x)+\delta)}$ but also with $\mathbb{P}\big[A_\delta^{(n)}\big] \geq 1-\delta$ . We shall revisit this construction.

We present a coding scheme: If $x^{(n)} \in A_\delta^{(n)}$, then code by indexing into this set using $1 + \lceil n(H+\delta)\rceil$ bits. If $x^{(n)} \notin A_\delta^{(n)}$ , we then code with $1 + \lceil n \log|\mathcal{X}|\rceil$ bits.

We now write out the expectation of the length of the codeword.

$$\mathbb{E}\big[l(x^{(n)}\big] = \sum_{x^{(n)}} p(x^{(n)}) l(x^{(n)} \leq \sum_{x^{(n)} \in A_\delta^{(n)}} p(x^{(n)})(2 + n(H+\delta)) + \sum_{x^{(n)} \notin A_\delta^{(n)}} p(x^{(n)})(2 + n\log|\mathcal{X}|)$$

$$\leq 2 + n(H+\delta) + n\log|\mathcal{X}|(1 - \mathbb{P}(A_\delta^n)$$

$$\leq 2 + n(H+\delta) + n\delta\log|\mathcal{X}| \equiv n(H+\epsilon) \text{ where } \epsilon = \frac{2}{n} + \delta(1+\log|\mathcal{X}|)$$

We now consider how to build $A_\delta^{(n)}$, often called a typical set. Let:

$$A_\delta^{(n)} = \{(x_1\cdots x_n) | H(x^{(n)}) - \delta \leq -\frac{1}{n}\log p(x_1\cdots x_n)\} \leq H(x)+\delta\}.$$

Clearly this is well defined.

**Claim 9.2** *If $X_1 \cdots X_n \rightsquigarrow p$ , then*

$$-\frac{1}{n}\log p(x_1, c\ldots x_n) \to H(X)$$

*as $n \to \infty$*

**Proof:** This essentially holds by the weak law of large numbers.

$$\frac{1}{n}\log p(x_1\cdots x_n) = -\frac{1}{n}\sum_i \log p(x_i) \to \mathbb{E}_{x\rightsquigarrow p} - \log p(x) = H(x)$$

■

The finite sample version holds by Hoeffding's inequality. If $p(x) \geq p_{min}$, then $0 \leq -\log p(x) \leq \log p_{min} \triangleq \gamma$ and as a result :

$$\mathbb{P}\big[\frac{1}{n}|\sum_i - \log p(x_i) - H(X)| \geq \epsilon\big] \leq 2e^{\frac{-2n\epsilon^2}{\gamma^2}}$$

or with probability $1-\delta$

$$|\frac{1}{n}\sum_i - \log(p(x_i)) - H(x)| \leq \sqrt{\frac{\gamma^2}{2n}\log(\frac{2}{\delta})}$$

Hence we simply set $A_{\epsilon,\delta}^{(n)}$ with $\epsilon = \sqrt{\frac{\gamma^2}{2n}\log(\frac{2}{\delta})}$ and we have $\mathbb{P}\big[A_{\epsilon,\delta}^{(n)}\big] \geq 1-\delta$

Hence, we have proven the achievability part of the Source Coding Theorem. Recapping, we use the Law of large numbers that most of the probability mass concentrates on a few sequences characterized by the entropy. We use the code that assigns short sequences to these sequences in the typical set and long sequences to the rest that are not in the typical set.

We make a few remarks about the above statement and proof. This is an asymptotic statement and where the asymptotics kick in depends on the distribution of p. (We note that we have a dependence on $\gamma = \log p_{min}$).

This code is non-singular for each $n$, but is extension is not necessarily uniquely decodable. Indeed the code may be completely different for each n since the typical sequences may change. Essentially we have constructed a sequence of codes (one for each n) that achieves the Shannon Rate. This code requires maintaining look up tables exponential in $n$ so this is not necessarily the most practical code construction. ■

### 9.2.1 Optimality and Kraft and McMillan Inequalities

We'll see that this construction using the typical set is optional and this has a few consequences. You cannot build a typical set with "better" properties. If you give me a set with good coverage then it will be roughly $2^{nH(x)+\epsilon}$ in size. To show this we need the following two lemmas.

**Lemma 9.3** Kraft's Inequality: *If a prefix free code has lengths* $(l_1, \cdots, l_A)$ *and is D-ary, then*

$$\sum_{i=1}^{A} D^{-l_i} \leq 1$$

**Proof:** Any prefix code can be thought of as a D-ary tree with code words on the leaves as shown in the diagram. No internal nodes can correspond to code words because that would violate the prefix property. If depth is $l_{\max}$, then if a symbol appears at depth $l_i$ , it annihilates $D^{l_{\max}-l_i}$ leaves so $\sum_i D^{l_{\max}-l_i} \leq D^{l_{\max}}$ where $D^{l_{\max}}$ is the maximum number of leaves at depth $l_{\max}$. ■

We also present an alternate proof: **Proof:** Drop 1 unit of dust at the root of the tree and split it evenly at each split, i.e send half down the left branch and half down the right branch at every split. If a leaf is at depth $l_i$, it accumulates $D^{-l_i}$ dust and by conservation since we only added one unit at the beginning at the root, necessarily $\sum_i D^{-l_i} \leq 1$

The other direction uses a similar idea. If $\sum_i D^{-l_i} \leq 1$ then build a D-ary tree of depth $l_{\max}$ and put the symbols at the correct levels. ■

**Lemma 9.4** McMillan's Inequality *If a uniquely decodable code has lengths* $(l_1, \cdots, l_A)$ *and is D-ary, then*

$$\sum_{i=1}^{A} D^{-l_i} \leq 1$$

**Proof:** Since we have already shown that we can construct a uniquely decodable code given $l_i$ with $\sum_{i=1}^{A} D^{-l_i} \leq 1$, and prefix-free codes $\subseteq$uniquely decodable codes, we have already shown one direction of the McMillan inequality.

For the other direction, we first clarify some notation. If $X$ is a symbol then $C(X)$ is the code word and $l(C(X))$ is the length. If $X^k$ is a k-sequence , then $C_k(X^k)$ is the codeword $C(X_1)\cdots C(X_k)$ with length $l(C_k(X_k))$. Hence

$$l(C_k(X_k)) = \sum_i l(C_k(X_i))$$

We also know that the number of codewords of length l is $n_l \leq D^l$ by uniqueness. We also note that this is not true for non-singular codes. As a result, we have

$$(\sum_x D^{-l(C(x))})^k = (\sum_{x_1} D^{-l(C(x_1))})(\cdots)(\sum_{x_k} D^{-l(C(x_k))}) =$$

$$= \sum_{x_1} \cdots \sum_{x_k} D^{-l(C(x_1))} \cdots D^{-l(C(x_k))} = \sum_{x^k} D^{-l(C_k(x^k))} = \sum_{l=1}^{kl_{\max}} n_l D^{-l} \leq \sum_{l=1}^{kl_{\max}} D^l D^{-l} = kl_{\max}$$

Hence $\sum_x D^{-lC(x))} \leq (kl_{max})^{\frac{1}{k}}$ and this holds for any $k$. Since the left hand side does not depend on $k$, it holds in the limit and since $\lim_{k\to\infty}(kl)^{\frac{1}{k}} \Rightarrow 1$, we conclude that

$$\sum_x D^{-l(x)} \leq 1$$

■

### 9.2.2 Achievability of Uniquely Decodable Codes

We find the D-adic distribution $r$, $r_i = \frac{D^{-l_i}}{\sum_j D^{-l_j}}$ minimizing

$$D(p\|r) - \log(\sum D^{-l_i}) \geq 0, D(p\|r) = \sum p_i \log(\frac{p_i \sum D^{-l_j}}{D^{-l_i}})$$

The choice is $l_i = \log_D \frac{1}{p_i}$ This however may not be integral so we round up to $l_i = \left\lceil \log_D \frac{1}{p_i} \right\rceil$ and we satisfy MacMillan's inequality since

$$\sum_i D^{-\left\lceil \log_D \frac{1}{p_i} \right\rceil} \leq \sum_i D^{-\log_D \frac{1}{p_i}} = \sum_i p_i = 1.$$

Hence the length is clearly $H(x) \leq L \leq H(x) + 1$

## 9.3 Shannon Source Coding Theorem: Necessity

**Theorem 9.5** *Shannon Source Coding Theorem (Necessity) : Any uniquely decodable coding scheme using D-ary codes has*

$$\mathbb{E}_p l_i(x) \geq H_p(x)$$

**Proof:** Consider the convex minimization problem:

$$\min_{l_i} \sum_i p_i l_i \text{ subject to } \sum_i D^{-l_i} \leq 1$$

with Lagrangians:

$$\mathcal{L} = \sum_i p_i l_i + \lambda(\sum_i D^{-l_i} - 1)$$

and derivative

$$\frac{\partial \mathcal{L}}{\partial l_i} = p_i - \lambda D^{-l_i} \log D = 0 \Rightarrow D^{-l_i} = \frac{p_i}{\lambda \log D}$$

Using complementary slackness, we know that the constraint should be tight so since $\lambda^*$ must be positive for the above to make sense.

$$\sum_i D^{-l_i} = \sum_i \frac{p_i}{\lambda \log D} = 1 \Rightarrow \lambda = \frac{1}{\log D}$$

or

$$D^{-l_i} = p_i \Rightarrow l_i = \log_D \frac{1}{p_i}$$

This concludes the proof of this theorem. ∎