COMP2610/COMP6261 Tutorial 3 Solutions*
Semester 2, 2018

Robert C. Williamson

August 20, 2018

1. (a) We know that X is a binomial with parameters n, p. Thus,

$$\begin{aligned} p(X \geq n-1) &= p(X = n-1) + p(X = n) \\ &= n \cdot p^{n-1} \cdot (1-p) + p^n. \end{aligned}$$

   (b) By Markov's inequality,

$$p(X \geq n-1) \leq \frac{\mathbb{E}[X]}{n-1} = \frac{n}{n-1} \cdot p.$$

   (c) When $n = 2$,

$$p(X \geq 1) = 2 \cdot p \cdot (1-p) + p^2 = p \cdot (2-p).$$

   The bound from Markov's inequality is

$$p(X \geq 1) \leq 2 \cdot p.$$

   The difference between the two is

$$2 \cdot p - (2 \cdot p - p^2) = p^2.$$

   Thus, for the Markov bound to be within 0.01 of the exact probability, we will need $p \leq 0.1$.

2. (a) We need to code all sequences of length 100 with three or less 1's. There is 1 sequence with zero 1's. There are 100 sequences with one 1 (one sequence for each possible position in which the 1 appears.) Similarly, there are $\binom{100}{2}$ and $\binom{100}{3}$ sequences with two and three 1's respectively. In total we have

$$1 + 100 + \binom{100}{2} + \binom{100}{3} = 166,751.$$

   If we want a uniform code over this number of elements, we need $\lceil \log_2(166,751) \rceil = 18$

   (b) We need to find the probability of observing more than 3 1's. Let K be the number of 1's observed. Then

$$\begin{aligned} P(K > 3) &= 1 - P(K \leq 3) = 1 - P(K = 0) - P(K = 1) - P(K = 2) - P(K = 3) \\ &= 1 - 0.995^{100} + 100 \times 0.995^{99} \times 0.005 \\ &+ \binom{100}{2} \times 0.995^{98} \times 0.005^2 + \binom{100}{3} \times 0.995^{97} \times 0.005^3 \\ &= 0.00167 \end{aligned}$$

   (c) The number of 1's, K, is a binomial random variable with probability of success 0.995. It has expectation $\mathbb{E}[K] = 100 \times 0.005 = 0.5$ and variance $\mathbb{V}[K] = 100 \times 0.995 \times 0.005 = 0.4975$. By Chebyshev's inequality we have

$$\begin{aligned} P(K \geq 4) &\leq P(|K - 0.5| \geq 3.5) = P(|K - \mathbb{E}[K]| \geq 3.5) \\ &\leq \frac{\mathbb{V}[K]}{3.5^2} = \frac{0.4975}{3.5^2} = 0.0406 \end{aligned}$$

   This is larger than the probability we found in part (b), it is off by a factor of around 20.

*Based in part on solutions by Avraham Ruderman for the 2012 version of the course.

3. (a) Using the definition of entropy

$$H(X) = 0.4 \times \log \frac{1}{0.4} + 0.6 \times \log \frac{1}{0.6} = 0.971$$

(b) i. Recall that

$$T_{N\beta} := \{\mathbf{x} : | - \frac{1}{N} \log_2 P(\mathbf{x}) - H(X)| < \beta\}$$

so we are looking for k (the number of ones) such that

$$0.871 = H(X) - 0.1 < -\frac{1}{N} \log_2 P(k) < H(X) + 0.1 = 1.071.$$

Referring to the table we see this is the case for values of k ranging from 11 to 19. That is, the typical set consists of sequences with between 11 and 19 1s.

ii. Summing the corresponding entries in the table (rows 11-19 in the second column) of the table we see

$$P(\mathbf{x} \in T_{N\beta}) = 0.043410 + 0.075967 + \ldots + 0.044203 = 0.936247$$

iii. The number of elements in the typical set is

$$4457400 + 5200300 + \ldots + 177100 = 26,366,510.$$

iv. By definition, S_δ is the smallest set of sequences such that $P(S_\delta) \geq 1 - \delta$. In particular, you always want to throw in the highest probability elements in order to get the smallest set. The highest probability sequences are the ones with the most 1s. If we start adding up probabilities from the third column of the table starting from the bottom (more 1s), we see that $P(K \geq 19) = 0.073564$ while $P(K \geq 18) = 0.15355$ so we need to add $\left\lceil \frac{(0.1-0.073564)}{0.079986} \times 480,700 \right\rceil = 158,876$ elements with 18 1s to the set of sequences with more than 18 1s. This gives us a total of

$$1 + 25 + 300 + 2,300 + 12,650 + 53,130 + 177,100 + 158,876 = 404,382$$

elements in S_δ.

v. The essential bit content is simply $\log_2 |S_\delta| = 18.625$.

4. (a) This is incorrect. Certainly it is true that $\frac{1}{N}H_\delta(X^N) \geq 0$, as the essential bit content is nonnegative. But this does not mean we can compress down to zero bits per outcome. If we were able to show that $\frac{1}{N}H_\delta(X^N) \leq \epsilon'$, for arbitrary ϵ', on the other hand, we would be able to make such a statement. Of course, we cannot show such a thing, because we know that the asymptotic fraction converges to the entropy $H(X)$.

(b) This is correct. The source coding theorem assumes that we are coding blocks created from extended ensembles. By definition this involves performing independent trials. When there is dependence amongst outcomes in a block, the theorem no longer holds.

5. (a) There are $52 - 20 = 32$ cards left, but only 13 distinct values, namely, $A_V = \{A, 2, 3, \ldots, 10, J, K, Q\}$. A uniform code over these 13 states takes $\lceil \log_2 13 \rceil = 4$ bits to code.

(b) $P(V)$ is given by:

$$\begin{aligned}
p(A) &= \frac{1}{32} \text{ since all that remains is } A\clubsuit \\
p(K) &= \frac{1}{16} \text{ since all that remains is } K\heartsuit, K\clubsuit \\
p(2) = p(3) = p(4) = p(5) &= \frac{1}{16} \text{ since all that remains are cards from } \clubsuit, \diamondsuit \\
p(J) = p(Q) = p(6) = p(7) = p(8) = p(9) = p(10) &= \frac{3}{32} \text{ since all that was removed were cards from } \spadesuit.
\end{aligned}$$

(c) We remove a subset so that S_δ is the smallest subset such that $p(v \in S_\delta) \geq 1 - \delta$:

δ	S_δ	H_δ (bits)
0	S_0	$\log_2 13 = 3.70$
$1/16$	$S_0 - \{A\}$	$\log_2 12 \approx 3.59$
$1/2$	$S_0 - \{A, K, 2, \ldots, 5, J\}$	$\log_2 6 \approx 2.59$

Here, S_0 is the set of all cards that we start with.

(d) We are looking for all the elements such that:

$$T_{1,0.3} = \{v : |-\log_2 p(v) - H(V)| < 0.3\}.$$

The entropy can be computed as:

$$H(V) = -\sum p_i \log p_i,$$

where $\mathbf{p} = (p_1, \ldots, p_32) = (1/32, 1/16, 1/16, 1/16, 1/16, 1/16, 3/32, \ldots, 3/32)$. Since $H(V) \approx 3.65$, this is equivalent to finding all v that satisfy:

$$\begin{aligned} H(V) - 0.3 < -\log_2 p(v) &< H(V) + 0.3 \\ 2^{-H(V)-0.3} < p(v) &< 2^{-H(V)+0.3} \\ 2^{-3.95} < p(v) &< 2^{-3.35} \\ 0.0647 < p(v) &< 0.0981. \end{aligned}$$

From 1(b) we see that those elements having $p(v) = 1/16 = 0.0625$ do not satisfy the above inequality, whereas those elements having $p(v) = 3/32 \approx 0.0938$ do. Hence:

$$T_{1,0.3} = \{J, Q, 6, 7, 8, 9, 10\}.$$

(e) No, it is not possible to do so. Using the source coding theorem (SCT), for large N, we see that $H(V) - \delta < \frac{1}{N} H_\delta(V^N)$. Hence $\frac{1}{N} H_\delta(V^N) > H(V) - \delta$. Substituting the corresponding values we have: $\frac{1}{N} H_\delta(V^N) > 3.65 - (1/16)$. Therefore we cannot have $\frac{1}{N} H_\delta(V^N) = 2$.

(f) Yes, the specified lengths satisfy Kraft's inequality: $3 \times 2^{-3} + 10 \times 2^{-4} = 1$. For instance the three face cards (J, K, Q) could be coded as $001, 010, 011$ and the ten non-face cards $(A, 2, 3, \ldots, 10)$ could be coded as $1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 0000, 0001$.

6. (a) The code C_1 is not prefix (0 is a prefix to 01) nor uniquely decodable (0110101 could decompose into 0 1101 01 or 01 10101). The code C_2 is prefix free and thus uniquely decodable. The code C_3 is not uniquely decodable (00 can be 00 or 0 0) and clearly not prefix free.

(b) $C_1' = \{0, 10, 1110, 11110\}$, $C_2' = C_2$. For C_3' we need a code a with lengths 1,1,2, and 2. However, this cannot be done as the only codewords of length one are 0 and 1 and one of these will be a prefix to any other codeword.

(c) No. The Kraft inequality tells us that for any prefix code

$$\sum_i 2^{-l_i} \leq 1$$

but for the given code lengths

$$\sum_i 2^{-l_i} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + 3 \times \frac{1}{16} = \frac{17}{16} > 1.$$

(d) Yes, these lengths satisfy the Kraft inequality. For instance $C = \{0, 100, 101, 1100, 1101, 1110\}$.

7. (a) We have

$$\begin{aligned} H(X) &= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{31}{128}\log\frac{31}{128} - \frac{1}{128}\log\frac{1}{128} \\ &= 1.55. \end{aligned}$$

(b) The expected code length is

$$L(C,X) = \frac{1}{2}\times 1 + \frac{1}{4}\times 2 + \frac{31}{128}\times 3 + \frac{1}{128}\times 3 = \frac{7}{4}.$$

(c) The code lengths for X are

$$\lceil \log_2 \frac{1}{1/2} \rceil = 1,\ \lceil \log_2 \frac{1}{1/4} \rceil = 2,\ \lceil \log_2 \frac{31}{1/128} \rceil = 3,\ \text{and}\ \lceil \log_2 \frac{1}{1/128} \rceil = 7.$$

An example of a prefix Shannon code for X would be:

$$C_S = \{0, 10, 110, 1110001\}$$

The expected code length would be

$$L(C_S,X) = \frac{1}{2}\times 1 + \frac{1}{4}\times 2 + \frac{31}{128}\times 3 + \frac{1}{128}\times 7 = 1.78125.$$

(d) We have

$$q_1 = 2^{-1} = \frac{1}{2},\ q_2 = 2^{-2} = \frac{1}{4},\ q_3 = 2^{-3} = \frac{1}{8},\ q_4 = 2^{-3} = \frac{1}{8},$$

(And $Z = 1$)

(e) By the definition of $D(\mathbf{p}||\mathbf{q})$ we have

$$\begin{aligned}
D(\mathbf{p}||\mathbf{q}) &= \sum_{i=1}^{4} p_i \log \frac{p_i}{q_i} \\
&= \frac{1}{2}\times \log_2 \frac{1/2}{1/2} + \frac{1}{4}\times \log_2 \frac{1/4}{1/4} + \frac{31}{128}\times \log_2 \frac{31/128}{1/8} + \frac{1}{128}\times \log_2 \frac{1/128}{1/8} \\
&= \frac{31}{128}\times \log_2 \frac{31}{16} + \frac{1}{128}\times 4 \\
&= 0.200.
\end{aligned}$$

So we have $D(\mathbf{p}||\mathbf{q}) = L(C,X) - H(X)$ as we would expect. We also note that $L(C_S,X)$ is greater (i.e. the code is worse) than C.

(f) The steps of Huffman coding would be:

- from set of symbols $\{x_1, x_2, x_3, x_4\}$ with probabilities $\{1/2, 1/4, 31/128, 1/128\}$, merge the two least likely symbols x_3 and x_4. The new meta-symbol x_3x_4 has probability $1/4$.
- from set of symbols $\{x_1, x_2, x_3x_4\}$ with probabilities $\{1/2, 1/4, 1/4\}$, merge the two least likely symbols x_2 and x_3x_4. The new meta-symbol $x_2x_3x_4$ has probability $1/2$.
- from set of symbols $\{x_1, x_2x_3x_4\}$ with probabilities $\{1/2, 1/2\}$, merge the two least likely symbols x_1 and $x_2x_3x_4$. The new meta-symbol $x_1x_2x_3x_4$ has probability 1, so we stop.

We then assign a bit for each merge step above. This is summarised below. We then read off the resulting codes by tracing the path from the final meta-symbol to each original symbol. This gives the code $C = \{0, 10, 110, 111\}$. (Note, we could equally derive $C = C_H$ depending on how we labelled the penultimate merge operation.)

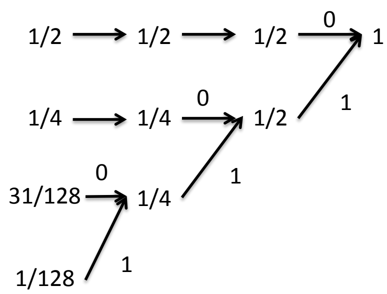

Figure 1: Huffman code.