Statistical Learning Theory, Exercise 2

Michael Hirsch
January 13, 2015

Source Coding

1. Compute or estimate the number of codewords you will need for this encoding scheme.

   This can be computed directly by calculating the number of pixel sequences that contain at most 3 black pixels.

   $$\binom{100}{0} + \binom{100}{1} + \binom{100}{2} + \binom{100}{3} = 1 + 100 + 4950 + 161700 = 166751$$

   □

2. What are your options for reducing these space requirements?

   Assuming that each sequence was stored in binary and that each occurrence of black or white was stored using 1 bit, for example, the 10 sequence consisting of "W,W,W,W,W,W,W,W,B,W" encoded as 0000000010, then we can instead encode n-tuples. For example:

   $$White, White \rightarrow 0$$
   $$Black, White \rightarrow 10$$
   $$White, Black \rightarrow 100$$
   $$Black, Black \rightarrow 111$$

   So our original sequence can be encoded as 0000100, instead of 0000000010, saving us 3 bits. This is just one example, and there are definitely more efficient ways of doing this for sequences of length 100. Actually, in the case (albeit very rare) of a string of 10 black pixels, this compression algorithm is *less* effective, requiring 30 bits to encode. However, this will be better in most cases.

   □

3. Bound the probability that this encoding scheme will encounter an untabulated sequence.

   If we take $Black = 1$ and $White = 0$ then $E[X_i] = 0.005$ and $Var[X_i] = (0.005)(0.995) = 0.004975$ for each random variable X_i

   We look at the random variable S_n which is the sum of precisely n X_i random variables. We will not have codes for sequences where $S_{100} \geq 4$. However, Chebyshev's inequality has $|S_n - nE[X]| \geq \epsilon$ so we must take $\epsilon = 3.5$, since $nE[X] = 100(0.005) = 0.5$

   $$Pr\{S_{100} \geq 4\} \leq \frac{100(0.005)(0.995)}{3.5^2} \approx 0.0406$$

   This is about 23 times larger than the actual value of 0.0017

   □