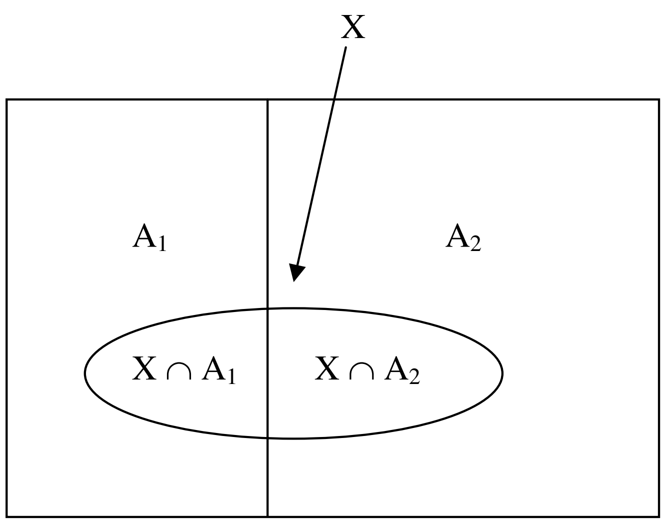
EE 121 - Introduction to Digital Communications
2008 Midterm Solutions

April 3, 2008

1. (a) Let X be an event. The total probability law says $\Pr(X) = \sum_i \Pr(X \cap A_i)$ where the A_i are exhaustive and mutually exclusive events, i.e. $\Pr(\cup_i A_i) = 1$ and $\Pr(A_i \cap A_j) = 0$ for all $i \neq j$. For two events A_1 and A_2 a Venn diagram illustrating this law is shown below.

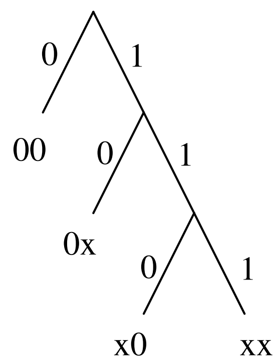


2. The mapping between the input signal and the signal transmitted over the channel is broken down into two serial mappings, the first converts the input signal into a compressed bit stream representing the data, the second converts the compressed bit stream into the signal to be transmitted over the channel. Some advantages are a simplified understanding of the communication system; the flexibility to swap sources without having to redesign the entire system; the flexibility to swap channels without having to redesign the source. Also, the separation is known to incur no performance loss.

3. (a) The average length per source symbol of the codeword generated, $\overline{L}$. In terms of the probability of the source taking on symbol i denoted p_i, and the length of the codeword associated with this symbol l_i, the average length per source symbol is $\overline{L} = \sum_i p_i l_i$.

(b) No, it only depends on the probability distribution.

(c) For block length 1 an optimal code is to assign 0 to symbol 0 and 1 to symbol 1. The average length is then 1. For block length 2 we use the Huffman coding algorithm to find the optimal code. The alphabet is $\{00, 0x, x0, xx\}$ with these symbols occurring with probabilities $\{(1-p)^2, p(1-p), p(1-p), p^2\}$. For p very small the algorithm will generate codewords $\{0, 10, 110, 111\}$.



The average length per source symbol is then $((1-p)^2 + 2p(1-p) + 3p(1-p) + 3p^2)/2$, which is approximately 1/2 for small p. For block length 3, the average length per source symbol will be approximately 1/3. This performance gain will not continue indefinitely, the average length per source symbol will approach the entropy of the source which $-p\log_2 p - (1-p)\log_2(1-p)$.

4. (a) No coded symbol, is a prefix is of another coded symbol. Prefix-free guarantees unique decodability of the code.

(b) Assuming $n > 0$, attach $\lfloor \log_2 n \rfloor$ zeros to the front of the binary representation of the integer. This prefix tells the decoder how long the binary representation of the integer will be. The binary representation requires roughly $\log_n$ bits. So this scheme requires roughly $2\log_2 n$ bits to code the integer n.

(c) Let N denote the number of self transitions from the 0 state to the 0 state, before a transition from 0 to 1 occurs.

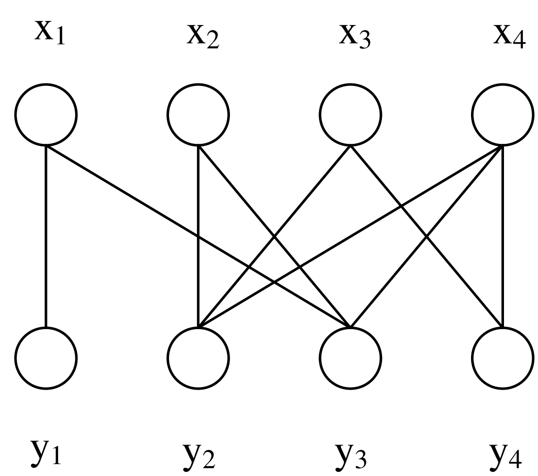
$$\begin{aligned}\mathbb{E}N &= \sum_{i=0}^{\infty} i \Pr(N = i) \\ &= \sum_{i=0}^{\infty} i\alpha(1-\alpha)^i \\ &= \frac{1}{\alpha} - 1.\end{aligned}$$

Then the expected time the chain stays in state 0 once it gets into state 0 is $1/\alpha$. Same answer for state 1 as the chain is symmetric.

(d) We expect to be coding integers of the order $1/\alpha$. This requires roughly $2\alpha \log_2(1/\alpha)$ bits per source symbol as opposed to 1 bit per source symbol. The compression rate is then one on this.

(e) The entropy rate is $\alpha \log_2(1/\alpha) - (1-\alpha)\log_2(1-\alpha)$. As $\alpha \to 0$ the entropy rate of the source is approximately $\alpha \log_2(1/\alpha)$. The average number of bits/symbol required for the run length code of part (c) is approximately $2\alpha \log_2(1/\alpha)$. As $\alpha \to \infty$ both quantities tend to zero, but the runlength code is roughly twice the entropy rate.

5. (a)



(b) Only one. There is only one seed y_1, the algorithm subtracts y_1 from y_3, but this does not generate another seed, hence the algorithm stalls at this point.

(c) All four. After decoding x_1 and subtracting it from y_3, we can subtract y_4 from y_2 to retrieve x_2, then subtract x_1 and x_2 from y_3 to retrieve x_4. Finally we subtract x_4 from y_4 to retrieve x_3.

(d) By enumeration, six of the 16 coefficient patterns produce a y_5 from which the iterative decoding algorithm can retrieve all four bits. Thus the probability is 3/8.

6. (a) The data rate $R = \log_2 M$. Thus there are $M = 2^R$ points in the constellation. The spacing between constellation points is $2A/(M-1) = 2A/(2^R-1)$, so the probability of error is $2Q(\sqrt{\text{SNR}}/(2^R-1))$ conditioned on a center point being transmitted and $Q(\text{SNR}/(2^R-1))$ conditioned on a boundary point being transmitted. Thus the probability of error is

$$2(1-2^{-R})Q(\sqrt{\text{SNR}}/(2^R-1))$$

At high SNR this is approximately $2(1-2^{-R})\exp(-\text{SNR}/2(2^R-1)^2)$. At high SNR the data rate required to maintain an error probability of p_0 will also be large and so $p_0 \approx$

$2\exp(-\text{SNR}/2^{2R+1})$ meaning the maximum achievable data rate will be about

$$R = \frac{1}{2}\left(\log_2 \text{SNR} - \log_2 \log \frac{2}{p_0} - 1\right)$$

(b) 6 dB (a factor of 4)

(c) Even with just $M = 2$, we still may not be able to meet the target error probability.

(d) $R = 1/n$. The ML detector chooses $-\sqrt{A}$ if $\sum_{m=1}^{n} y[m] < 0$ and $+\sqrt{A}$ if $\sum_{m=1}^{n} y[m] > 0$. Thus an error occurs if $\sum_{i=1}^{n} w[i] > n\sqrt(A)$ and the error probability is then $Q(\sqrt{n\text{SNR}}) = Q(\sqrt{\text{SNR}/R})$. Thus $R = \text{SNR}/(Q^{-1}(p_0))^2$.

(e) 3 dB (a factor of two).

(f) Scenario (2), as the return on the investment is greater, that is, for a given increase in SNR, we get a more significant increase in rate.