MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
Department of Mechanical Engineering

6.050J/2.110J **Information and Entropy** **Spring 2004**

Problem Set 6 Solutions

Solution to Problem 1: Prime Massachusetts Madness

Table 6–1 summarizes our knowledge at the beginning of the problem.

Probability	Value	Corresponding Statement
$p(RB \mid D)$	0.30	30% of the Democrats received Republican ballots.
$p(DB \mid D)$	0.70	[70% of the Democrats received Democrat ballots]
$p(DB \mid R)$	0.20	20% of the Republicans received Democrat ballots
$p(RB \mid R)$	0.80	[80% of the Republicans received Republican ballots]
$p(DB \mid I)$	0.50	50% of the Independents received Democrat ballots
$p(RB \mid I)$	0.50	50% of the Independents received Republican ballots

Table 6–1: Our Knowledge of the Probabilities

Solution to Problem 1, part a.

The desired probabilities are $p(DB \mid D) = p(RB \mid R) = 1$ and $p(RB \mid D) = p(DB \mid R) = 0$. The actual probabilities are given in Table 6–1.

Solution to Problem 1, part b.

Here we are given the information that $p(D) = 0.3$ and $p(R) = 0.2$ in Newton. We can conclude that $p(I) = 0.5$. From Bayes' theorem we know that $p(RB, X) = p(RB)p(X \mid RB) = p(X)p(RB \mid X)$. Since we have the last two numbers for all three cases, we can calculate our answer as:

$$
\begin{aligned}
p(total) &= 0.3 \times 0.3 + 0.2 \times 0.8 + 0.5 \times 0.5 = 0.5 \\
p(RB, D) &= \frac{0.3 \times 0.3}{p(total)} = 0.18 \\
p(RB, R) &= \frac{0.2 \times 0.8}{p(total)} = 0.32 \\
p(RB, I) &= \frac{0.5 \times 0.5}{p(total)} = 0.50
\end{aligned}
$$

Solution to Problem 1, part c.

Similarly we can calculate the numbers for Cambridge:

$$
\begin{aligned}
p(total) &= 0.6 \times 0.3 + 0.2 \times 0.8 + 0.2 \times 0.5 = 0.44 \\
p(RB, D) &= \frac{0.6 \times 0.3}{p(total)} = 0.41 \\
p(RB, R) &= \frac{0.2 \times 0.8}{p(total)} = 0.36 \\
p(RB, I) &= \frac{0.2 \times 0.5}{p(total)} = 0.23
\end{aligned}
$$

Solution to Problem 2: Special Orders Don't Upset Us

Solution to Problem 2, part a.

This is a noisy channel with the same probabilities for mixing up Z and B. Channel capacity is defined as the maximum mutual information (for any possible input probability) times the rate W. The rate of error is $\epsilon = 0.10$. Channel capacity for this channel is given by Equation 6.26.

$$
\begin{aligned}
C &= M_{max} W \\
&= 1 - \epsilon \log_2\left(\frac{1}{\epsilon}\right) - (1-\epsilon)\log_2\left(\frac{1}{(1-\epsilon)}\right) \\
&= 1 - 0.10 \log_2\left(\frac{1}{0.10}\right) - (0.90)\log_2\left(\frac{1}{0.90}\right) \\
&= 0.5158 \text{ bits/second} \qquad (6\text{–}1)
\end{aligned}
$$

$$(6\text{–}2)$$

Solution to Problem 2, part b.

Information per symbol is defined as

$$
\begin{aligned}
I &= p \log_2\left(\frac{1}{p}\right) + (1-p)\log_2\left(\frac{1}{1-p}\right) \\
&= 0.05 \log_2\left(\frac{1}{0.05}\right) + (0.95)\log_2\left(\frac{1}{0.95}\right) \\
&= 0.2234 \text{ bits} \qquad (6\text{–}3)
\end{aligned}
$$

$$(6\text{–}4)$$

Solution to Problem 2, part c.

Since the information per symbol, I, is less than the channel capacity, C, we can send our information reliably.

Solution to Problem 2, part d.

Our maximum rate is the channel capacity (bits/second) divided by the information content per order (bits/order), to give us a maximum order rate per second of 2.31.

Solution to Problem 2, part e.

Table 6–2 and 6–3 shown the solution to this problem.

Character	Code
Burger-Burger	B
Zucchini-Burger	ZB
Burger-Zucchini	ZZB
Zucchini-Zucchini	ZZZ

Table 6–2: Huffman code for Buzz

Order	Probability	Bits	Average # of Bits
Burger-Burger	0.9025	1	0.9025
Zucchini-Burger	0.0475	2	0.095
Burger-Zucchini	0.0475	3	0.1425
Zucchini-Zucchini	0.0025	3	0.0075
Total	1.00		1.1475

Table 6–3: Information content of Huffman code for Buzz

The channel capacity is 0.5158 bits/second, or 1.0316 bits every two seconds. This is lower than our needed capacity, so the scheme will not work.

Solution to Problem 3: Groceries a Dodo

Solution to Problem 3, part a.

In the first problem, orders could be lost when ITM packets from the refrigerator are dropped. Since the server does not send confirmation, the refrigerator does not know if an order has been lost.

In the second problem, if the final EDO packet is lost, the server will wait indefinitely for it, and cause it to hang.

Solution to Problem 3, part b.

A refrigerator might have its timeout set much shorter than the server can respond. In this case, the server seems particularly slow, and so the refrigerator send 10,000 ITM packets, each containing a request for a single bottle of ketchup, before the server acknowledged the request.

Solution to Problem 3, part c.

There are multiple solutions to this problem.