Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
Department of Mechanical Engineering

6.050J/2.110J **Information and Entropy** **Spring 2006**

Issued: March 7, 2006

Problem Set 5 Solutions

Due: March 10, 2006

Solution to Problem 1: Give Peas a Chance

Strains

Solution to Problem 1, part a.

If he had wanted to evaluate all possible cross interaction between all te traits he would have needed $2^7 = 128$ instances. You can think of each characteristic as a bit taking any of two values and there are 7 characteristics. However, since he only wanted to evaluate each characteristic separately he only needed 14 strains.

Hybrids

Solution to Problem 1, part b.

The table below shows the predictions of Blending theory face to face with the observations of Mendel.

	Blending Theory	**Mendel Observation**
smooth seeds	1%	100%
golf ball seeds	98%	0%
wrinkled seeds	1%	0%
Total	100%	100%

First Generation from the Hybrids

	Experiment 2 Shape of Albumen			
Trial	Smooth	Wrinkled	Percentage of Smooth Seeds	Percentage of Wrinkled Seeds
1	45	12	79%	21%
2	27	8	77%	23%
3	24	7	77%	23%
4	19	10	66%	34%
5	32	11	74%	26%
6	26	6	81%	19%
TOTAL	173	54	76%	24%

Table 5–2: Table for the second set of experiments, extended with the total and the proportion of wrinkled and smooth seeds.

Solution to Problem 1, part c.

Table 5–2, shows the result of computing the proportion of wrinkled seeds.

Solution to Problem 1, part d.

Table 5–2 shows noticeable fluctuations in the proportion of wrinkled seeds around a value of 25%. Considering experimental errors, a good choice for the probabilities of each trait is: $P(seed = wrinkled) = 1/4$ and $P(seed = smooth) = 3/4$.

Mendel did many such experiments, for this characteristic and also six others. Some of his findings are in Table 5–3.

Experiment	**Trait**	**count**	**Trait**	**count**
Shape of ripe seeds	*smooth*	5474	*wrinkled*	1850
Seed color	*green*	2001	*yellow*	6022
Length of stem	*long*	787	*short*	277
Color of the seed-coats	*gray-brown*	705	*white*	224

Table 5–3: Summary of results for the first generation from the hybrids. This table is extracted from Gregor Mendel's original work.

Solution to Problem 1, part e.

The proportion of 3 to 1 is still apparent from table 5–3. Here we can see is the essential difference between statistics and probability. The more data we have the more we expect statistics to reproduce the probabilistic model. However nature samples from this probabilistic model randomly, and this random sampling will introduce certain departures from the expected ratio.

Second Generation from the Hybrids

Solution to Problem 1, part f.

The problem statement gives us already $P(r_2, D_1) = 1/8$. From part (d), we know that $P(D_1) = 3/4$ and by Bayes' rule,

$$P(r_2, D_1) = P(r_2|D_1) \cdot P(D_1) \implies P(r_2|D_1) = \frac{1/8}{3/4} = 1/6$$

Consequently, since only two outcomes are possible: $P(D_2|D_1) = 1 - P(r_2|D_1) = 5/6$. We know that if the parent exhibited the recessive trait so will its offspring, therefore: $P(r_2|r_1) = 1$, and once again, since only two outcomes are possible: $P(D_2|r_1) = 0$.

Solution to Problem 1, part g.

Since D_1 and r_1 are a partition $P(D_2) = P(D_2, D_1) + P(D_2, r_1) = P(D_2|D_1)P(D_1) + 0 = 5/6 \times 3/4 = 5/8$.

Solution to Problem 1, part h.

We are being asked to compute the probability: $P(D_1|D_2)$. Applying Bayes' rule in the other direction:

$$P(D_1|D_2) = \frac{P(D_1, D_2)}{P(D_2)} = 1.$$

Solution to Problem 2: Huffman Coding

Solution to Problem 2, part a.

In order to encode 7 characters we would need 3 bits. That is $2^3 = 8$ codewords, and a total of $3 \times 33 = 99$ bits to transmit the sentence.

Solution to Problem 2, part b.

Table 5–4 lists the calculation of the average information per symbol. Here we calculate an average of 2.1252 bits per symbol, or 49 bits.

Character	#	Frequency	$\log_2\left(\frac{1}{p_i}\right)$	$p_i \log_2\left(\frac{1}{p_i}\right)$
d	10	30.30%	1.7226	0.5220
space	5	15.15%	2.7322	0.4112
a	5	15.15%	2.7322	0.4112
d	5	15.15%	2.7322	0.4112
u	5	15.15%	2.7322	0.4112
o	2	6.06%	4.0445	0.2451
y	1	3.03%	5.0445	0.1528
Total	33	100.00%		2.5647

Table 5–4: Frequency distribution of characters in "yubba dubba dubba dubba dubba doo"

Solution to Problem 2, part c.

See Table 5–4.

Solution to Problem 2, part d.

A possible code is derived below and listed in Table 5–5.

Start: (b='NA' $p = 0.3030$) (space='NA' $p = 0.1515$) (a='NA' $p = 0.1515$)(d='NA' $p = 0.1515$)(u='NA' $p = 0.1515$) (o='NA' $p = 0.0606$)(y='NA' $p = 0.0303$)

Step 1: (b='NA' $p = 0.3030$) (space='NA' $p = 0.1515$) (a='NA' $p = 0.1515$)(d='NA' $p = 0.1515$)(u='NA' $p = 0.1515$) (o='0' y='1' $p = 0.0909$)

Step 2: (b='NA' $p = 0.3030$) (u='0' o='10' y='11' $p = 0.2424$) (space='NA' $p = 0.1515$) (a='NA' $p = 0.1515$)(d='NA' $p = 0.1515$)

Step 3: (b='NA' $p = 0.3030$) (a='0' d='1' $p = 0.3030$) (u='0' o='10' y='11' $p = 0.2424$) (space='NA' $p = 0.1515$)

Step 4: (u='00' o='010' y='011' space='1'$p = 0.3939$) (b='NA' $p = 0.3030$) (a='0' d='1' $p = 0.3030$)

Step 5: (b='0' a='10' d='11' $p = 0.6060$) (u='00' o='010' y='011' space='1'$p = 0.3939$)

Final code: (b='00' a='010' d='011' u='100' o='1010' y='1011' space='11'$p = 1$)

Character	Code
b	00
space	11
a	010
d	011
u	100
o	1010
y	1011

Table 5–5: Huffman code for "yubba dubba dubba dubba dubba doo"

Solution to Problem 2, part e.

When the sequence is encoded using the codebook derived in part d...

i. See Table 5–6.

Character	# of Characters	Bits per Character	Bits Needed
d	10	2	20
space	5	2	10
a	5	3	15
d	5	3	15
u	5	3	15
d	2	4	8
u	1	4	4
Total	33		87

Table 5–6: Huffman code for "yubba dubba dubba dubba dubba doo"

ii. The fixed length code requires 99 bits, whereas Huffman coding requires 87 bits. So we find that the Huffman code does a better job than the fixed length code.

iii. This number compares extremely well with the information content of 84.63 bits for the message as a whole.

Solution to Problem 2, part f.

The original message is 33 bytes long, and with LZW we know from Problem Set 3 we can encode the message using LZW in 21 bytes, with 18 additional entries in the dictionary. Thus we need 18+9=27 different dictionary entries (do not forget the 7 characters and the start and stop signals), for a total of 5 bits per byte. Thus we can compact the message down to $21 \times 5 = 105$ bits. Straight encoding needs 99 bits, and Huffman encoding needs 87 bits. Thus Huffman encoding does the best job of compacting the material (assuming we do not need to transmit the codebook).