MITOCW | MIT6_004S17_01-02-12-04_300k

This problems presents Huffman Encoding which produces variable-length encodings based on the probabilities of the occurrence of each different choice.

More likely choices will end up with shorter encodings than less likely choices.

The goal of this problem is to produce a Huffman code to encode student choices of majors.

There are a total of 4 majors, and each has a probability associated with it.

To produce a Huffman encoding, one begins with the 2 choices with lowest probability.

In this example, major 6-7, has a probability of 0.06, and major 6-1, has a probability of 0.09.

Since these are the two lowest probabilities, these two choices are selected as the starting point for building our encoding tree.

The root node that combines the two choices, has a probability equal to the sum of the leaf nodes, or 0.15 in this example.

We then label one side of this tree with a 0 and the other with a 1.

The next step is to find the next two smallest probabilities out of the remaining set of probabilities where majors 6-1 and 6-7 have been replaced by node A which has probability 0.15.

In this case, our lowest probabilities are 0.15 (which is the probability of node A) and 0.41 (which is the probability of major 6-3).

So we create a new node B that merges nodes A and 6-3.

This new node has probability 0.56.

Again we label the two branches, one with a 0 and the other with a 1.

We now repeat this process one last time with the only two remaining choices which are now node B and major 6-2.

This means that we should make a new node C that merges node B and major 6-2.

Note that the probability of this node is 1.0 because we have reached the top of our tree.

Our final step is to label these last two branches.

Now that all the branches are labeled, we can traverse the tree from the root node to each leaf node in order to identify the encoding that has been assigned to the major associated with that leaf node.

We find that for major 6-1 the encoding is 101.

For major 6-2, we end up with a 1-bit encoding of 0.

Next we traverse the tree to identify the last two encodings and find that for major 6-3 the encoding 11 has been assigned, and for major 6-7 the encoding 100 has been assigned.

These encodings make sense because we expect the major with the highest probability, in this case major 6-2 to end up with the shortest encoding.

The next highest probability major is 6-3 so it ends up with the second shortest encoding, and so on.

We just saw that the encodings resulting from this Huffman encoding tree are: 101 for major 6-1, a 0 for major 6-2, 11 for major 6-3, and 100 for major 6-7.

Note that the Huffman encoding tree for this problem could have also been drawn like this.

These two trees are identical in structure and result in the same encodings for the four majors.

Furthermore, a Huffman tree can result in more than one valid encoding.

The only constraint in labeling the edges is that from each node, there is both a 0 branch and a 1 branch but there are no constraints about which side has to be labeled 0 and which side 1.

So for example, we could have chosen to label the left side of the B node with a 1 and the right side with a 0 instead of the way we originally labeled them.

Note, however, that this would result in a different but also valid Huffman encoding.

In this case the encoding for major 6-1 is 111, major 6-2 remains 0, major 6-3 becomes 10, and major 6-7 becomes 110.

As long as one maintains consistency across the selected tree, a valid Huffman encoding is produced.

We now add one more column to our table which gives p * log2(1/p) for each of the majors.

Using this information we can calculate the entropy which is the average amount of information contained in each message.

This is calculated by taking the sum of p * log2(1/p) for all choices of majors.

For this problem, the entropy is 1.6.

We can now also calculate the average bits per major of the encodings that we have identified.

This is calculated by multiplying the number of bits in each encoding times the probability of that major.

Recall that our encoding for major 6-1 was 111, for major 6-2 it was 0, for major 6-3 it was 10, and finally for major 6-7 it was 110.

This means that the average bits per major is 3 times 0.09 plus 1 times 0.44 plus 2 times 0.41 plus 3 times 0.06 = 0.27 plus 0.44 plus 0.82 plus 0.18 which equals 1.71.

Note that this is slightly larger than the entropy which is 1.6.

This occurs because while Huffman encoding is an efficient encoding which gets us most of the way there, there are still some inefficiencies present in Huffman encoding because it is only encoding one major at a time rather than also considering the probabilities associated with seeing a particular sequence of majors in a message that conveys the major selected for a large number of students.