SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF BIOMEDICAL ENGINEERING

19BMB302 - BIOMEDICAL SIGNAL PROCESSING

III YEAR/ V SEMESTER

Unit V : DATA REDUCTION TECHNIQUES

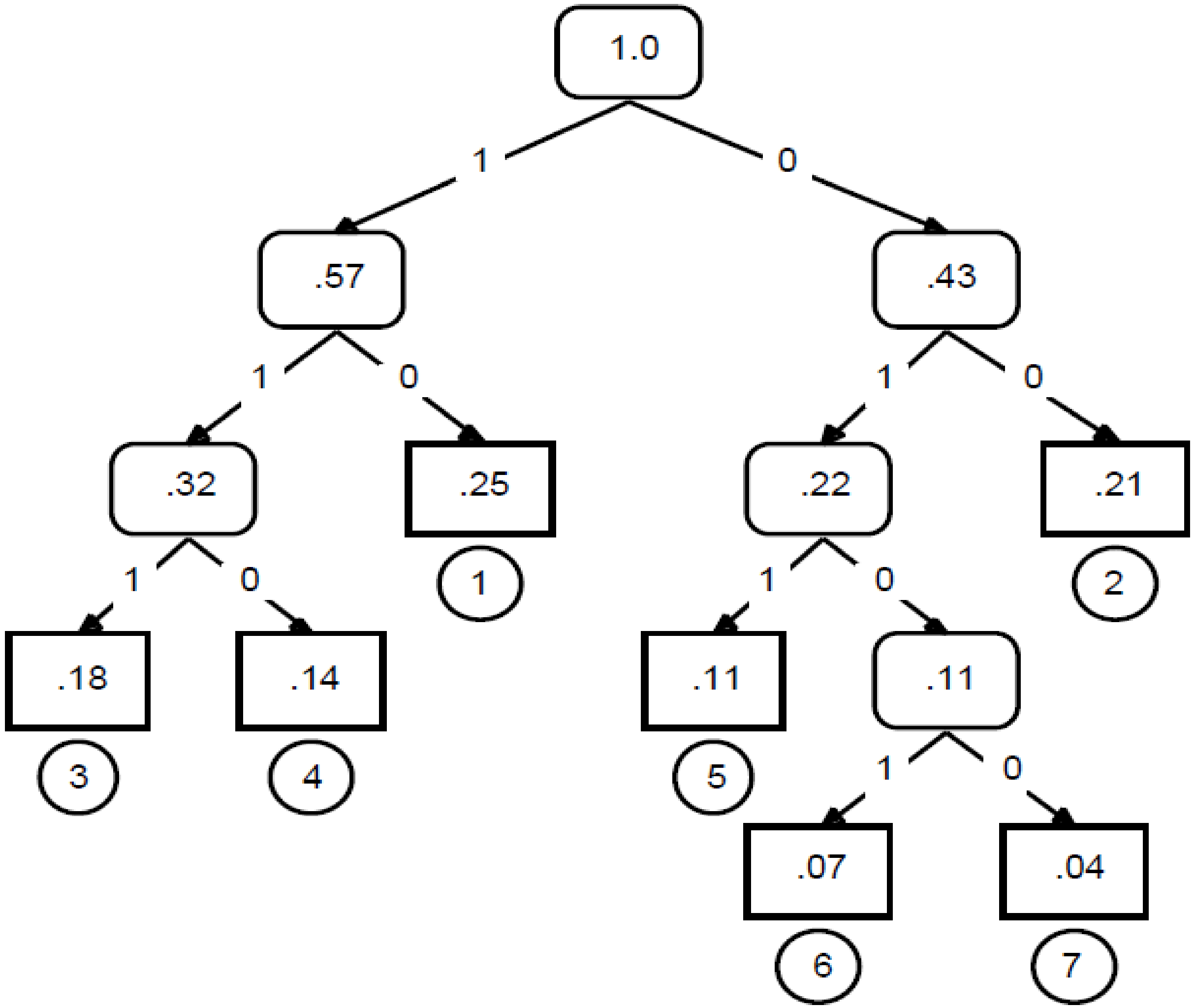
- Turning point algorithm
- AZTEC algorithm
- CORTES algorithm
- Fan algorithm
- Huffman algorithm

Huffman algorithm

- Huffman coding exploits the fact that discrete amplitudes of quantized signal do not occur with equal probability It assigns variable-length code words to a given quantized data sequence according to their frequency of occurrence.
- Data that occur frequently are assigned short
- Figure 1 illustrates the principles of Huffman coding.
- As an example, assume that we wish to transmit the set of 28 data points {1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 6, 6, 7}
- The set consists of seven distinct quantized levels, or *symbols*. For each symbol, *Si*, we calculate its probability of occurrence *Pi* by dividing its frequency of occurrence by 28, the total number of data points.
- Consequently, the construction of a Huffman code for this set begins with seven nodes, one associated with each *Pi*. At each step we sort the *Pi* list in descending order, breaking the ties arbitrarily.
- The two nodes with smallest probability, *Pi* and *Pj*, are merged into a new node with probability *Pi* + *Pj*.
- This process continues until the probability list contains a single value, 1.0, as shown in Figure

S_i	Lists of P_i						
1	.25	.25	.25	.32	.43	.57	1.0
2	.21	.21	.22	.25	.32	.43	
3	.18	.18	.21	.22	.25		
4	.14	.14	.18	.21			
5	.11	.11	.14				
6	.07	.11					
7	.04						

- The process of merging nodes produces a binary tree as in Figure
- When we merge two nodes with probability *Pi* + *Pj*, we create a parent node with two children represented by *Pi* and *Pj*.
- The root of the tree has probability 1.0. We obtain the Huffman code of the symbols by traversing down the tree, assigning 1 to the left child and 0 to the right child.
- The resulting code words have the *prefix property* (i.e., no code word is a proper prefix of any other code word).
- This property ensures that a coded message is uniquely decodable without the need for lookahead.


1.0
1
0
.57
.43
1
0
1
0
.32
.25
1
.22
.21
2
1
0
1
0
.18
3
.14
4
.11
5
.11
1
0
.07
6
.04
7

- Figure summarizes the results and shows the Huffman codes for the seven symbols.
- We enter these code word mappings into a translation table and use the table to pad the appropriate code word into the output bit stream in the reduction process.
- The reduction ratio of Huffman coding depends on the distribution of the source symbols.
- In our example, the original data requires three bits to represent the seven quantized levels.

Symbols, S_i	3-bit binary code	Probability of occurrence, P_i	Huffman code
1	001	0.25	10
2	010	0.21	00
3	011	0.18	111
4	100	0.14	110
5	101	0.11	011
6	110	0.07	0101
7	111	0.04	0100

- *li* represents the length of Huffman code for the symbols. This value is 2.65 in our example, resulting in an expected reduction ratio of 3:2.65.
- The reconstruction process begins at the root of the tree.
- If bit 1 is received, we traverse down the left branch, otherwise the right branch.
- We continue traversing until we reach a node with no child. We then output the symbol corresponding to this node and begin traversal from the root again.
- The reconstruction process of Huffman coding perfectly recovers the original data.
- Therefore it is a lossless algorithm. However, a transmission error of a single bit may result in more than one decoding error.
- This propagation of transmission error is a consequence of all algorithms that produce variable-length code words.

Thank You!