CHAPTER 3

REPRESENTATION AND ANALYSIS OF ELECTROCARDIOGRAM

3.1 INTRODUCTION

The continuing proliferation of computerized ECG processing systems along with the increased feature performance requirements and demand for lower cost medical care have mandated reliable, accurate ECG data compression techniques. Having preprocessed the electrocardiograms by removing the artifact and noise, such as baseline wander and powerline interference, the electrocardiograms are represented in a suitable form to achieve data compression and then for analysis. This chapter deals with a novel method of ECG representation using data structures.

3.2 TREE AS DATA STRUCTURE

Tree is a type of data structure that is used to represent related data items in a particular ordered form. In general tree is a finite set of elements in which one element is called the root of the tree and remaining elements are partitioned into disjoint subsets, each of which is known as a subtree.

Figure 3.1 illustrates a sample tree structure. The prominent dots, in the figure, denote the nodes of the tree. The number of subtrees of a node is called its degree. Also, the nodes that have a zero degree or no subtrees are said to be the leaf nodes or terminal nodes of the tree, and the nodes
Figure 3.1 Tree representation
that have a non-zero degree or a non-zero number of subtree are said to be
the non-leaf nodes or non-terminal nodes. In Figure 3.1 the degree of node
11 is therefore 3, and the nodes (1,2,3,4,7,9,10) are the leaf nodes, and all
remaining are non-leaf nodes. The degree of the tree is the degree of that
node which has maximum number of sub-trees. The degree of the tree in
Figure 3.1 is therefore three.

By analogy, the relationship between the nodes in a tree can be
considered as equivalent to the father-son relationship in a family setup.
That is subtrees of the node ‘X’ are said to be its sons. In other words, node
X is the father of all its subtrees. Descendant of a node are all the nodes
that belong to all the subtrees of that node. Therefore, in a tree all nodes
are descendants of the root node. The ancestors of a node are all nodes that
lie along the path from the root to that node. In Figure 3.1 descendants of
node 6 are nodes (1,2,3,4,5) and ancestors of node 6 are nodes (8,11). The
level of a node is defined as follows. The root of a tree has a level zero and
the level of any other node in the tree is one more than the level of its
father. The depth of a tree is the maximum level of any node in the tree. In
Figure 3.1, the level of node 7 is 2, which is one more than the level of its
father node 8 and depth of the tree is 4.

Thus the tree is a two-dimensional data structure with its degree
and level, which can be used to represent waveforms. The tree
representation of a waveform provides a linear description of succession of
peaks and valleys in the waveform and also a self embedded structure of the
waveform.

Waveforms can be represented using trees in three different ways,
namely Relational tree representation, Skeletal tree representation and
Complete tree representation (Roger W. Ehrich et al., 1976 ; Yao-Chou
Cheng et al., 1985).
3.2.1 Relational Tree representation

Construction of a relational tree is briefly discussed below. First, a root node is created at the lowest value of the waveform. This node is labeled with highest peak in the waveform. The lowest valley divides the waveform into two sub-waveforms. For each sub-waveform, a node is created with the lowest value in the sub-waveform and labeled with highest peak in the waveform. If no valley could be found in the sub-waveform, leaf nodes are created corresponding to the peak in the sub-waveform and are labeled with that peak. This process is repeated until all peaks and valleys in the waveform are covered. Figure 3.2 illustrates this method.

The deepest valley in the waveform is $V_1$. The root is formed at $V_1$ and is named as $P_4$, the highest peak in the waveform. The valley $V_1$ divides the waveform into two sub-waveforms, first and second. $P_1$ is the highest peak in the first sub-waveform and no valley could be found in the first sub-waveform. So a leaf node is created corresponding to the first sub-waveform and is labeled with $P_1$, the highest peak in the first sub-waveform. The second sub-waveform contains peaks $P_2$, $P_3$, $P_4$ and valleys $V_2$, $V_3$. The valley $V_3$ is the lowest in the second sub-waveform. A node is created at $V_3$ and is labeled with $P_4$, the highest peak in the second sub-waveform. The second sub-waveform further divides the second sub-waveform into two more sub-waveforms, third and fourth. The fourth sub-waveform contains peak $P_4$ and no valleys. So a leaf node is created corresponding to fourth sub-waveform and is labeled as $P_4$. The third sub-waveform has $V_2$ as the lowest peak. A node is created at $V_2$ and is labeled with $P_2$, since $P_2$ is the highest peak in third sub-waveform. The valley $V_2$ divides the third sub-waveform into two more sub-waveforms, fifth and sixth. The fifth and sixth sub-waveforms contain peaks $P_2$ and $P_3$ respectively and no valleys. So leaf nodes are created corresponding to these peaks, originating at the node corresponding to valley $V_2$ and are labeled accordingly.
Figure 3.2 Relational tree representation
Properties of Relational tree representation

1. The leaves of a tree give a linear description of peaks in the waveform.
2. Each non-leaf node of the tree corresponds to a valley in the waveform.

The relational tree as a whole describes the relative position of peaks in a waveform. Quantitative information like amplitude and duration can be included by attaching attributes to the tree nodes.

3.2.2 Skeletal Tree Representation

For the construction of skeletal tree, the amplitude of the waveform is first quantized. Each pair of crossings enclosed by the waveform on a quantization level delineates an interval. For the first quantization level, the interval has the duration of the entire waveform. A node is created for this interval and is designated as a root of the tree. From the interval corresponding to the current node to the interval on the next quantization level above, tree nodes are created and assigned as the son nodes of the current node. This recursive procedure is repeated until no further crossings can be found. Figure 3.3 shows a waveform represented using skeletal tree with eight quantization levels.

Properties of Skeletal Tree Representation

1) The depth of a tree is equal to the number of quantization levels.
2) The leaves of the tree are linear description of peaks in that waveform.
3) The depth of a leaf node is equal to the amplitude of its corresponding peak on that waveform.
Figure 3.3 Skeletal tree representation
4) The nearest common predecessor of two nodes represents the valley in that waveform.

A disadvantage of this representation is that small peaks below the resolution of quantization are missed. Peak $P_4$ in Figure 3.3 is missed as a result of quantization error. This can be overcome by increasing the number of quantization levels.

3.2.3 Complete Tree Representation

The complete tree representation is an extension of skeletal tree representation in the sense that the width information of peaks and valleys are added to the skeletal tree representation. With both quantization and sampling used, constructing a complete tree is like overlaying a grid on a waveform. The procedure is similar to the construction of a skeletal tree. The waveform is thus observed through a grid structure with vertical and horizontal grid lines. Here, in addition to nodes created for each interval enclosed by that waveform on a quantization level, nodes are also created for each point where the waveform cuts the vertical grid line between the current quantization level and the next quantization level. Figure 3.4 is an illustration of complete tree representation of a sinusoidal waveform.

The algorithm which is used to generate complete tree representation of a waveform is presented below.

The algorithm has two parts. The first part creates the root node of the first quantization level. The second part creates the sub-trees starting from the root recursively until the waveform is represented fully.
Figure 3.4 Complete tree representation of a sinusoidal waveform
Algorithm for generating a complete tree:

Input to the algorithm: The floating point numeric data samples in the form of an array $Y(n)$.
X-increment: A positive integer quantity.
Y-increment: A positive real quantity.
Requirement: $Y(0) = Y(N) = 0.0$
Output: A complete tree representation of the waveform

Function: Maketree

1) A root node is created at quantization level '0' at the midpoint of the interval (0-N).
2) Set current quantization level = 0.
3) Call the function subtree(0,n,quantization level)

Function: subtree(k,l,a)

1) Increment quantization level ‘a’ by Y-increment.
2) Create a new array $Y_h(j)=Y(i)$ such that $i$ is an integral multiple of $X$-increment.
3) Find the points $cut_1, cut_2, cut_3,..., cut_m$, such that the waveform cuts the current quantization level and the vertical line of the grid. That is $(Y(cut_j)<a)$ and $(Y(cut_{j+1})>a+Y$-increment) or $(Y(cut_j)>a)$ and $(Y(cut_{j+1})<a+Y$-increment). Also, $(Y(h(cut_j) \geq a)$ and $(Y(h(cut_j) \leq a+Y$-increment).
The quantity 'm' is always an even number.

4) For each pair of the cutting points, a node is created at the midpoint of the interval formed by the pair of cutting points. This newly formed node is the son node of the current node located at quantization level 'a' and is at the midpoint of interval (k,l). Each pair of cutting points forming an interval and their quantization level are then passed as arguments to the function subtree. The function is called recursively and further subtrees are formed. The process is repeated until no more cutting points could be found.

Tree representation is implemented using pointers. The structure of each of the tree node is as follows:

Tree node:  
- item1(integer type) : cutting point-1
- item2(integer type) : cutting point-2
- item3(integer type) : number of subtrees of the node
- item4(integer type) : postorder index of the node
- item5(real type) : quantization level of the node
- son[MAXSONS] : Array of pointers (tree node type) pointing to the son nodes of a node

### 3.3 REPRESENTATION OF ECG WAVEFORM USING COMPLETE TREE

As described in section 3.2.3, the complete tree is generated for the ECG waveform. Figure 3.5 shows the ECG Tree representation.
As per the complete tree generation algorithm, every node of the tree has five items as given in the previous section. Any node of the tree with both item1 and item2 of equal values is identified as leaf node or otherwise as non-leaf node. Item5 of a node gives the y-amplitude level of that node. Therefore for a leaf node, either item1 or item2 (both are equal) gives the x-magnitude and item5 gives the y-magnitude. In other words (item1,item5) of a leaf node represents the data sample of ECG at that position of leaf node.

For reconstruction of this ECG waveform, only leaf nodes are used.

The other items, namely item3 and item4 are useful for matching of any two waveforms (Lu, S.Y., 1985; Scott W. Shaw et al., 1990; Parthasarathy, N., et al., 1991). In this work, the matching of waveforms are not considered.

3.3.1 ECG data compression

The electrocardiograms of sampling rate 500 sps and 300 sps are used. The leaf nodes are extracted by using the method described in section 3.3. The results obtained by using 100 vertical grids as well as 50 vertical grids are given in the Table 3.1.

Table 3.1 Results of ECG data compression using complete tree representation

<table>
<thead>
<tr>
<th>Waveform</th>
<th>Parameter</th>
<th>No.of Vertical grids in the grid structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>ECG waveform of 500 sps</td>
<td>RMSE</td>
<td>0.015012</td>
</tr>
<tr>
<td></td>
<td>PRD</td>
<td>12.252349</td>
</tr>
<tr>
<td></td>
<td>Compression ratio</td>
<td>3.33:1</td>
</tr>
<tr>
<td>ECG waveform of 300 sps</td>
<td>RMSE</td>
<td>0.000689</td>
</tr>
<tr>
<td></td>
<td>PRD</td>
<td>2.62407</td>
</tr>
<tr>
<td></td>
<td>Compression ratio</td>
<td>2:1</td>
</tr>
</tbody>
</table>
The Relative Mean Square Error (RMSE) and the Percent RMS Difference (PRD) are calculated by using the equations shown below.

Relative Mean Square Error,
\[
RMSE = \frac{\sum_{i=1}^{N} (f(i) - \hat{f}(i))^2}{\sum_{i=1}^{N} f^2(i)}
\]  \hspace{1cm} (3.1)

Percent Root-mean-square Difference,
\[
PRD = \frac{\text{RMS error}}{\text{RMS}} \times 100 \%
\]  \hspace{1cm} (3.2)

where RMS error \( = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (f(i) - \hat{f}(i))^2} \) \hspace{1cm} (3.3)

\[
\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} f^2(i)}
\]  \hspace{1cm} (3.4)

and \( f(i) \) and \( \hat{f}(i) \) are the original and reconstructed ECG waveforms.

The compression ratio is expressed in terms of the number of bytes used in the original and compressed electrocardiograms.

From the Table 3.1, it is observed that percent RMS difference is less in the case of 100 grids than that of 50 grids, irrespective of the sampling frequency of the electrocardiograms.

The error due to compression of a typical ECG waveform with sampling rate of 300 sps is shown in Figure 3.6.
Figure 3.6 Comparison of compression errors
(a) Original ECG (300 sps) (b) ECG reconstructed from 100 leaf nodes (c) ECG reconstructed from 50 leaf nodes (d) and (e) compression error for waveforms (b) & (c) respectively
3.3.2 Measurements of amplitude and duration of ECG waveform

The lead LII ECG waveform is used for the standard basic measurements like PR interval, PT interval, RT interval, QT interval, P wave duration, QRS duration, T wave duration and R-S amplitude on ECG.

A cycle of electrocardiogram as shown in Figure 3.7 is fitted into the grid in PQRST complex order. For proper fitting of ECG into the grid, a standard QRS detector is used.

From the feature nodes, extracted from the electrocardiogram using complete tree algorithm, with 100 vertical grids, the interval and peak amplitudes are measured.

The algorithm for the measurements of interval and peak magnitude:

Input
1. Max nodes - maximum number of leaf nodes.
2. Node array - array of leaf nodes and each node consists of item1 and item5.

Output
Array of closed curves.

From the array of closed curves, the width and height information are calculated.

Table 3.2 shows comparison of values obtained from this method (ECG Tree) and the direct measurement from the ECG itself.
Figure 3.7 A cycle of typical electrocardiogram
Table 3.2  Comparison of measured ECG parameters

<table>
<thead>
<tr>
<th>Parameter measurements from</th>
<th>ECG waveforms of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal ECG (sec)</td>
</tr>
<tr>
<td>Tree structure</td>
<td></td>
</tr>
<tr>
<td>PR interval</td>
<td>0.14</td>
</tr>
<tr>
<td>QRS duration</td>
<td>0.08</td>
</tr>
<tr>
<td>QT interval</td>
<td>0.40</td>
</tr>
<tr>
<td>ST segment</td>
<td>0.15</td>
</tr>
<tr>
<td>Direct measurements by cursor</td>
<td></td>
</tr>
<tr>
<td>PR interval</td>
<td>0.15</td>
</tr>
<tr>
<td>QRS duration</td>
<td>0.08</td>
</tr>
<tr>
<td>QT interval</td>
<td>0.39</td>
</tr>
<tr>
<td>ST segment</td>
<td>0.14</td>
</tr>
<tr>
<td>Error between the above two measurements</td>
<td></td>
</tr>
<tr>
<td>PR interval</td>
<td>0.01</td>
</tr>
<tr>
<td>QRS duration</td>
<td>0</td>
</tr>
<tr>
<td>QT interval</td>
<td>0.01</td>
</tr>
<tr>
<td>ST segment</td>
<td>0.01</td>
</tr>
</tbody>
</table>
3.4 COMPLETE TREE REPRESENTATION OF ECG FOR FEATURE EXTRACTION

In the previous section 3.3 the ECG waveform is represented by complete tree and it is expected that the ECG waveform is reconstructed back. In this section, a complete tree algorithm is presented to extract feature (skeletal) nodes from the ECG waveform for classification of ECG waveforms or heart diseases based on ECG waveform abnormalities. The structural features in ECG waveform such as presence of Q-wave, ST elevation inverted T-wave, tall R and deep S can be extracted over the space spanning from post P-wave to post T-wave in the ECG waveform. For this an amplitude-time grid of size $15 \times 15$ is overlaid on the waveform. The 15 leaf nodes which are the vertical intersection of the grid structure with the ECG waveform are searched from the bottom to top and from right to left (Figure 3.8). These 15 leaf nodes are the structural features of the ECG extracted through $15 \times 15$ grid structure. The algorithm for constructing a complete tree is given below.

Algorithm for complete tree:

Input

1. Constant K is the number of Quantization levels.
2. Constant N is the number of sample points.
3. Waveform f(x)
4. Quantization level i.

Output

A complete tree T.
Figure 3.8 Feature extraction of ECG through
15 x 15 grid structure
Procedure

1. Move to the lowest quantization level
2. Set $i=1$
3. Move to the rightmost point of signal
4. Set $N$ as maximum.
5. Move from right to left and mark the points on the grid where the signal crosses the vertical grids.
6. Move to the next quantization level.
7. Set $i=2$
8. Move to the rightmost point of signal in the current quantization level.
9. Search for signal intersections with the vertical grid by moving from right to left.
10. Repeat procedure till the top most quantization level containing the signal is reached.

Using this algorithm, 15 feature nodes are extracted from the electrocardiograms of the heart diseases namely, Anteroseptal Myocardial infarction and Left Ventricular Hypertrophy for classification. In chapter 4 the neural network classifier for the classification is described.

3.4.1 Grid Compression

Sometimes the entire single cycle of the electrocardiogram cannot be properly fitted on the grid. If so, the extraction of the nodes in this case will produce less number of nodes, and hence valuable information regarding the ECG will be lost. In order to avoid this, two methods are implemented to extract complete information. In the case of the grid compression, the entire grid structure is compressed in order to fit the signal on the grid. This is done by finding the starting and the end points of the array consisting of the ECG data and the 15 x 15 grid structure is drawn with these points. Once this is done the nodes are extracted using the tree algorithm.
### 3.4.2 Signal Expansion

Another procedure that could be followed to avoid loss of information, in case the original signal does not completely fit in the grid, is called as Signal Expansion. In this case the signal or a part of the signal to be analyzed is ‘stretched’ along the entire original grid. The original grid is not altered but the signal is fitted into the grid structure. After this, the tree extraction algorithm is used to obtain the pattern vectors or the leaf nodes.

### 3.5 Equivalent Tree Representation of Electrocardiogram Signals Using Genetic Algorithm

The Genetic algorithm described in section 2.5 is used in the reduction of feature nodes extracted using complete-tree representation discussed in section 3.4. In this work, the 15 feature nodes (pattern vectors) extracted from the electrocardiograms are reduced to the optimized 10 nodes, and thus redundancies in the features are removed by the application of genetic algorithm.

From the pattern vectors of 15 nodes, a population is generated for the algorithm to work with. This population has five strings, each consisting of 10 nodes randomly chosen from the 15 nodes. This is the initialization of the population. The next step is to find the fitness value of the strings in the population. Since genetic algorithm is an optimization procedure, in order to obtain the optimized 10 node-string from the current population, a fitness function is defined as given below.

The fitness function,

\[ f_n = \sum_{i=1}^{10} (|y(i) - y(i-1)| + |y(i+1) - y(i)|) \]  

where \( f_n \) is the fitness value of \( n^{th} \) string in the current population.
This fitness function represents the summation of the vertical distances of the neighboring nodes of the target node in the string. The distance is calculated for each and every node in the string, and the fitness value is the summation of all such values. The fitness values for the five strings are calculated and the values are normalized. These normalized values are studied and depending on the values, 'Selection' is done, i.e., the offsprings for the strings are allocated. An offspring is allocated if the fitness value is in the range of 0.8 to 1.0.

The next very important step is 'Crossover'. In this case the probability of crossover is fixed at 0.8, i.e., four out of the five strings undergo crossover. The field to perform the crossover is fixed at 2, i.e., all the nodes from the third onwards up to the tenth are crossed over. This crossover is performed taking any two strings at a time and exchanging all the nodes after the field length.

Sometimes during the crossover some important nodes might be lost. To avoid loss of information, mutation is performed as the next step. It is done, with a certain probability which is very small. In this case, one node out of the entire population is considered and replaced with the best node.

The procedure of selection, crossover and mutation are performed until the termination criterion of the best fitness value is reached. Thus, the best nodes are obtained from the initial 15 nodes by genetic algorithm. This reduced number of 10 leaf nodes from the original 15 leaf nodes of the ECG waveform, obtained by complete tree representation, is named as the equivalent tree representation of ECG waveform. The validity of this equivalent tree is tested by a neural network classifier which is discussed in chapter 4.