Time-Recurrent HMM Decision Tree to Generate Alerts for Heart-Guard Wearable Computer

Swati Keskar, Rahul Banerjee

Birla Institute of Technology and Science, Pilani, India

Abstract

In this paper, we propose a time-recurrent decision tree in order to generate time bound alerts by the wearable computing system in cases where in the wearer is found to be in a medical state that deserves to be attended. This decision tree leads to generate alerts by picking up most likely state sequence, based on trained Hidden Markov Models (HMM) and acquired real-time signal. We present simulation results based on Physikalisch-Technische Bundesanstalt Database (PTBDB), available on physionet.org as it contains ECG as well as VCG data records from 294 subjects (healthy as well as having various heart diseases). The construction and learning of decision tree was carried out using Iterative Dichotomiser 3 (ID3) algorithm, which was found suitable for varying importance of different parameters for different classes of heart disease

1. Introduction

Wearable computers are being widely used in physiological/ECG monitoring. There are quite a few wearable monitoring systems or portable ambulatory systems designed by researchers. Most of these work on heart rate variability or arrhythmia. In some cases RR interval is within normal limits but other parameters show abnormality in the functioning of the heart which may go unnoticed. For early detection of signs we need to monitor the polarization/depolarization of atria and ventricles. Due to the constraints of computing power of wearable computing, we have to rely on relevant temporal and amplitude parameters. This signifies the importance of modeling and decision making. As ECG is a non-stationary signal, we modeled ECG using Hidden Markov Model (HMM) which is suitable for modeling such signals. VCG leads 'V_x', 'V_y', 'V_z' are used for modeling, which provide the same information as 12-leads of ECG.

Before constructing the tree, labeling of data needs to be done. Since the PTBDB data is classified in various disease classes but beats are not labeled, we have to follow Viterbi path to reach the particular class. For this purpose we divided the data available in each category into two parts, test data and training data. Eight learning curves were obtained representing seven classes of diseases and normal one. These curves are stored in a look up table. The most probable state sequence can then be obtained which gives maximum likelihood of detection of problem state. A time-recurrent HMM decision tree was constructed using ID3 algorithm, to generate different intensity level alerts depending upon the state sequence with other parameters. An example of dysrhythmia (arrhythmia) class and decision tree constructed for it is presented in this paper.

2. Related work

HMM and decision trees are vastly used in speech signal processing for long. Casey et al used probabilistic model of decision tree for character recognition [1]. Bahl et al developed a continuous speech recognition system by using decision trees [2]. A real-time recurrent word recognition system was proposed by Tony Robinson [3]. It was a hybrid system using connectionist model and Markov model. Foote J. further evolved this into time-recurrent decision tree probability model for HMM speech recognition [4]. A time series model, as combination of decision tree with Markov temporal structure was proposed by Jordan et al [5]. Wieben et al used filter bank features, fuzzy rules and decision trees for classification of Premature Ventricular complexes for classification [6]. No parallel model for ECG signal modeling and classification using our method is known to date.

3. System overview

An overview of the wearable computing system used to generate alerts using HMM and Decision Trees is as shown in Figure 1.

The system includes support for three sensor leads to acquire electrical activity of the heart, signal processing, feature extraction, Viterbi algorithm, classification and decision making unit to generate alert. While generating alert, current state vector was also considered with input

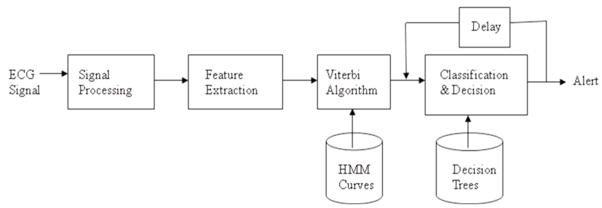

Figure 1. Block diagram for alert generation

vector. To extract features, a four seconds window was considered. In normal ECG signal, this window has four to five beats. Average values of certain parameters of these beats were extracted and were fed to Viterbi network. Viterbi algorithm uses these parameters to pick up the most likely state sequence, representing any of the class defined with the help of trained HMMs. Decision Trees were also trained to generate alerts according to the classification of signal using proper sequence of parameters. This alert is refreshed after every four seconds.

Alerts generated are of three types, having different intensity levels.

Level 1: First level alert is generated to give wearer a gentle warning to indicate some abnormality in the functioning of the heart.

Level 2: If problem is more severe then second level alert is generated for the patient as well as his kin and registered doctor. This type of alert is transmitted over a distance through mobile of the wearer using appropriate cellular wireless communication services like 2.5G, 3G, 3.5G, 4G etc. Radio frequency transmitter facilitates communication between wearable computing unit and paired mobile phone/smart phone/ mobile computing device of the wearer.

Level 3: Third intensity level is generated in case of emergency. Though there will be a local alert generated, the wearer may not in a position to act or respond. At this level, he definitely needs some immediate help. This calls for the provision for following types of specialized alerts in addition to the local cautionary alerts.

I: A severity information alert, which is sent in layman's language to next of his kin along with location information.

II: A detailed technical information based alert, which shall be sent along with patient's history metatag and location information to the registered medical specialist if any.

III: A mapping service based nearest trauma or emergency response center shall be identified by the system which shall be then communicated. This alert may contain preliminary information about patient in layman's language and compressed technical information about the condition of the patient with exact location coordinates.

4. Hidden markov model

The ECG signal is a complex, non-stationary signal. HMM is widely used for modeling of such signals. It can be treated as a time synchronized finite state machine having distinct states, which switch from one state to other at regular or semi regular interval. In our case, we have modeled the ECG signal using HMM technique [7]. Keeping in mind, the constraints and limitations of wearable unit, we focused on temporal parameters and peak amplitudes. This model has 10 input states, Q_1, R_1, S, J, ISO_1, T_p, P_p, ISO_2, Q_2 and R_2. Three output parameters RR interval (heart rate HR), QT interval and ST depression were observed to cover entire activity of heart. In this paper, we have added one more parameter QRS interval to construct the decision tree.

The PTBDB database is broadly classified into eight classes of heart diseases, including normal. For online monitoring eight HMMs were trained and learning curves stored as look up table. Viterbi algorithm was then used to find the most likely to path, using the state sequence, to arrive at the classification.

5. Viterbi learning algorithm

As said HMM have 10 states and four output parameters. This model was used to predict a heart disease from given input state sequence. Viterbi algorithm was used to align input sequence with most probable path [8]. For that purpose, it uses knowledge of the parameters of the HMM model and a particular output sequence. It then finds the state sequence that is most likely to have generated that output sequence. What if more than one state sequences produce the same output sequence? In our case, it was not a trivial concern since we had to generate alerts.

A network for Viterbi learning is shown in Figure 2. A generic lattice and trellis diagram for the complete PTBDB database was prepared.

The output parameters are O_1: RR interval, O_2: QT interval, O_3: QRS interval and O_4: ST Depression. RR interval is further used to find heart rate HR (inverse of RR interval) in beats per minutes (bpm) by projecting. All are temporal parameters. Eight classes of heart diseases are, A: Normal, B: Dysarythmia, C: Myocardial Infarction, D: Bundle branch block, E: Cardiomyopathy, F: Hypertrophy, G:Valvular heart disease and H: Myocarditis.

The probability of the most probable path ending in state '*k*' with observation '*i*' is found out by the recursive method given below.

$P_l\,(i,n) = B_l(i)\ max_k\,(P_k\,(j,\ n\text{-}1)*P_{kl})$

where

$B_l(i)$ = probability to observe element 'i' in state l

P_k *(j, n-1)* = probability of the most probable path ending at position n-1 in state '*k*' with element *j*

P_{kl}= probability of the transition from state '*l*' to state '*k*'

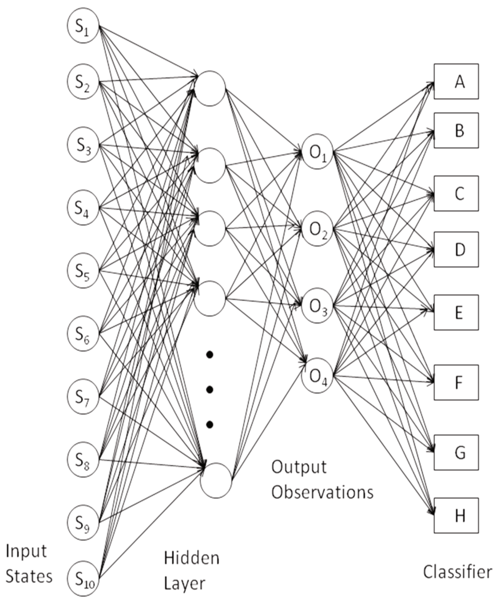

Figure 2. Network for Viterbi learning

6. Decision tree construction

A decision tree is a tree structure consisting of decision nodes and leaf nodes. At decision nodes, a test is performed on the parameter. Leaf nodes represent category. Depending on the result of the test, decision is taken for appropriate branch, decision node or leaf node. *Iterative Dichotomiser 3 (ID3)*, a well known iterative algorithm was used for tree construction and classification [9]. How to select small and appropriate tests (or questions) from large number of possibilities is trivial. The size of resulting tree also must be as short as possible.

There are three rules for constructing decision trees, splitting rule, stopping rule and labeling rule. In our case, we constructed tree in two stages. In first stage, a tree was constructed to test whether the parameter value is within normal range or not. In second stage, another tree was constructed to test the parameter value to generate alerts of various intensity levels.

Figure 3 shows a generic example of top down decision tree, constructed in first stage. Four parameters were tested, in random manner in given example. We constructed decision tree for each class of heart diseases, in which we chose proper root parameter, proper sequence of parameters to test and arrive at decision. These trees were stored as look up table as shown in Figure 1.

The classification task involved four parameters as stated earlier

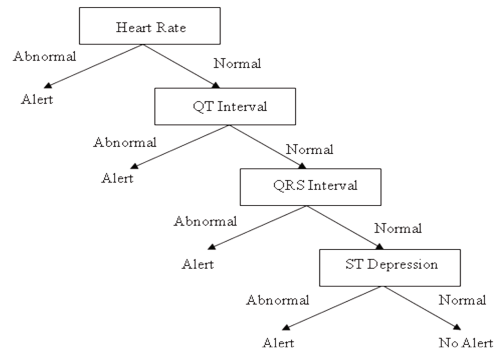

Figure 3. Decision Tree to generate alert

Heart rate HR : values [low, high, higher, very high]
QT interval: values [normal, long, short]
QRS interval: values [normal, short, long]
ST Depression: values [normal, elevated, depressed]

All class models were described by these parameters.

Splitting and labeling, based on these parameters, was carried out by Iterative Dichotomiser 3 algorithm (ID3). A training_set consisting of all four parameters was constructed. Due to space constraint, we chose to present a subset of this training_set as shown in Table 1 and constructed a simple tree for HR parameter.

The choice of questions or tests was crucial. We chose appropriate threshold levels to induct decision and leaf nodes as shown in Figure 4. *Entropy* and *Information gain* played an important role in construction and learning of tree. Splitting and labeling was done at the same time. Tree construction was terminated when all samples were classified

Table 1. Training_set for Heart Rate HR

Class of Dysarythmia	Qualitative Interpre-tation	Threshold Range for HR (bpm)	Alert	Level of Alert
Bradycardia	HR < *low*	< 60	Yes	level 1
Normal	*low* < HR< *high*	60 to 100	No	No Alert
Tachycardia	*high*<HR < *higher*	100 to 250	Yes	level 1
Flutter	*higher* <HR< *very high*	250 to 350	Yes	level 2
Fibrillation	HR > *very high*	> 350	Yes	level 3

*HR (in bpm) in normal case: low:*60 bpm; *high*:100 bpm; *higher*:250 bpm ; *very high*:350 bpm

7. Decision tree learning

Entropy and I*nformation gain* decided which node is to be followed next depending on outcome of the test. Samples for this were obtained from dysrhythmia class of PTBDB database. Let P_p represents number of positive samples and P_n negative samples. Then Entropy (measure of impurity) for all samples at root node is

$$Entropy(S) = - P_p \log_2 P_p - P_n \log_2 P_n$$

The same procedure was carried at all nodes by inducting two nodes at a time until all samples are classified and entropy became zero. Then information gain for these samples using the parameter was calculated

$$Gain(S, HR) = E(S) - \Sigma E \text{ (all child nodes)}$$

Figure 4 shows a simple decision tree, which is part of a complex tree constructed in second stage.
Figure 5 shows algorithm for it.

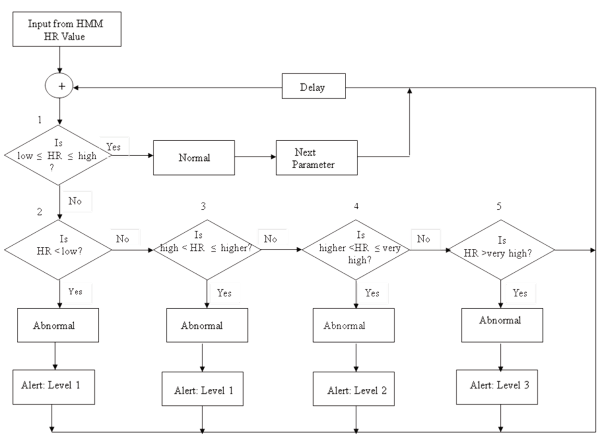

Figure 4. Decision tree for HR, based on RR interval

```
START    input :  HR value, CurrentAlarmStatus

         begin
         if  ((HR >=low) &&(HR <= high)) then
             { return  Normal
             Go to NextParameter
              Delay
              Go to START  }
         elseif  ( HR< low)  then
             { return Abnormal
             Alert : Level 1}
          elseif((HR >high) &&( HR<=higher)) then
             { return Abnormal
              Alert: Level 1 }
         elseif  ((HR> higher)&&  (HR<= very high))  then
             { return Abnormal
             Alert : Level 2 }
         elseif  ( HR > very high)  then
             { return Abnormal
           Alert : Level3 }
        end
     Delay
     Go to START
```

Figure 5. Algorithm to generate 3 level alerts

8. Conclusions and discussion

A real-time time-recurrent decision tree was constructed, which takes input from HMM. In this paper, we have shown that, training HMM and decision tree offline and storing them in database, helps to increase the efficiency and accuracy of the model at less computing power. We have shown construction and learning of decision tree, only for one disease. Accuracy achieved in classification of two classes, at first node in figure 4, was pretty high. As we moved from top to bottom accuracy changed remarkably. Same procedure was repeated for other diseases and appropriate parameters were chosen to generate alerts of three intensity levels. ID3 algorithm allowed construction of shortest and fastest tree. Cross-validation and training on other databases is to be carried out in the next phase to increase the accuracy in probabilistic inference process.

References

[1] Casey R, Nagy G. Decision Tree Design Using a Probabilistic Model. IEEE Trans on Information Theory. 1984;30(2):93-99.

[2] Bahl L, DeSouza P, Gopalakrishnan P, Nahamoo.D, Picheny M. Decision trees for phonological rules in continuous speech. Proc. ICASSP. 1991(46).185-188

[3] Robinson T. A Real-Time Recurrent Error Propagation Network Word Recognition System. Proc. ICASSP. 992:47. 617-620.

[4] Foote J. Decision-Tree Probability Modeling for HMM Speech Recognition. PhD Thesis. Brown University. 1991.

[5] Jordan M, Ghahramani Z, Saul L. Hidden Markov Decision Tree. Proc. Advances Neural Information Processing systems. 1996 (9).

[6] Wieben O, Afonso V, Tompkins W. Classification of premature ventricular complexes using filter bank features, induction of decision trees and a fuzzy rule-based system. Springer Link journal of Medical and Biological Engineering and Computing: 1999: 37(5): 560-565.

[7] Keskar S, Banerjee R. Modeling ECG Signal for Supervised Training Using HMM for BITS-Heart-Guard Wearable Computer. National Conference on Virtual And Intelligent Instrumentation. 2009. 90-95.

[8] Forney G Jr. The Viterbi Algorithm. Proc. IEEE, 1973:61(3).263-278.

[9] Mitchell T. Machine Learning. McGraw Hill 1997.

Address for correspondence:

Swati Keskar
EEE/Instrumentation Department
Birla Institute of Technology and Science,
Pilani, India 333031
swatikeskar@bits-pilani.ac.in