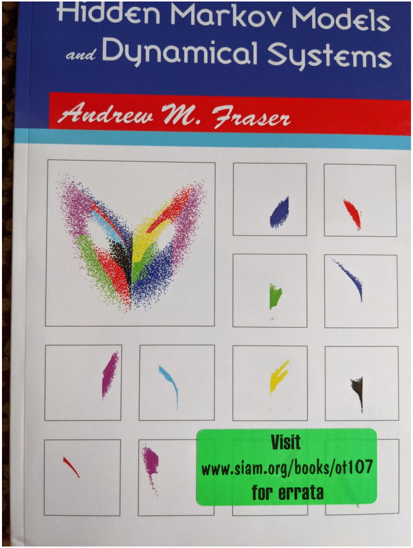
Using Discrete State Hidden Markov Models to Estimate Heart Rate

Andrew M. Fraser

SIAM DS23

2023-5-14

Book from Mini-Symposium at DS2001 on Hidden Markov Models



Discrete state dynamics

$$P(s[t+1] \mid s[t])$$

Simplest case: discrete observation models

$$P(y[t] \mid s[t])$$

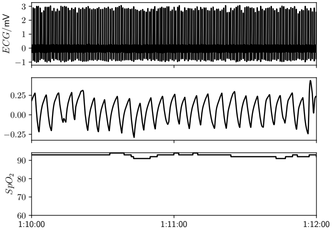
For ECG: Autoregressive observation models

$$P(y[t] \mid s[t], y[t-3], \cdots y[t-1])$$

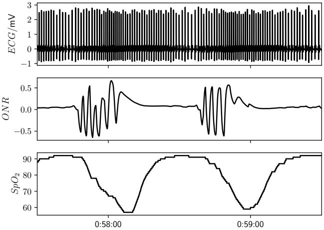
Goal of CINC 2000: Use ECG to Detect Apnea

Computers in Cardiology 2000 Challenge: Classify EKG

Normal



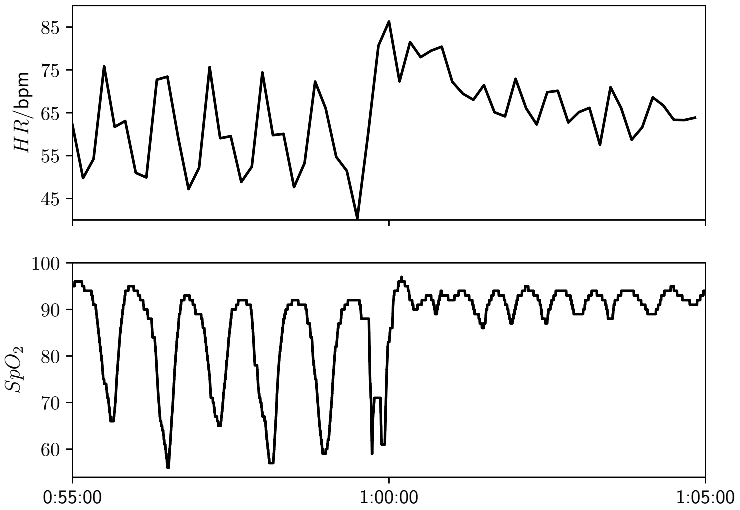
Apnea



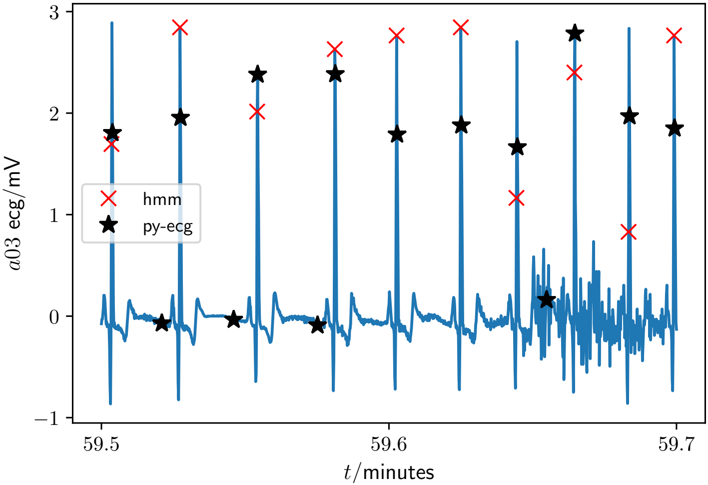
Intermediate Objectives:

Detect QRS Pattern → Estimate Heart Rate

See Apnea in Heart Rate

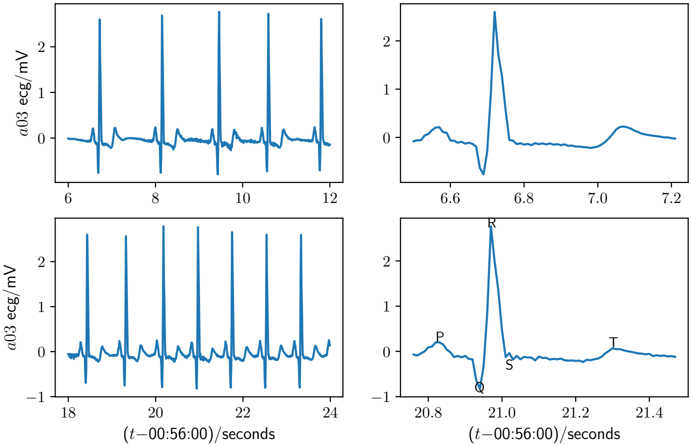


QRS From GitHub



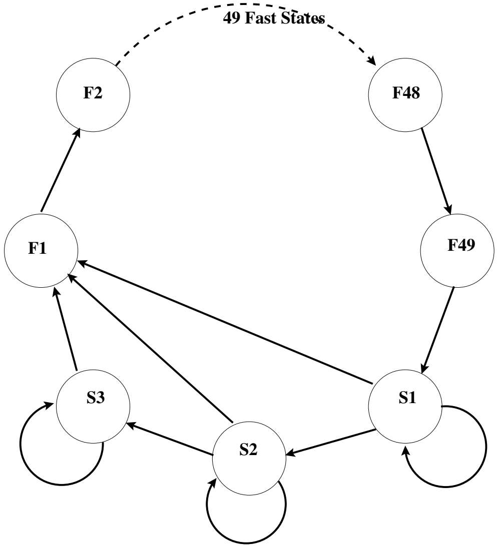
Results from https://github.com/berndporr/py-ecg-detectors aren't good enough.

Unvarying PQRST Duration



At different heart rates the shape and duration of the PQRST pattern doesn't change. Only the delay between the sequences changes.

HMM State Structure



Hidden Markov Model State Structure

- Loop of 52 discrete states
- A sequence of 49 *fast* states that don't branch, state n must transition to state $n + 1$ at each time step
- Three *slow* states accommodate heart rate variations. Each of the three branches to one of the following:
  - Itself
  - Its successor
  - The first fast state

The minimum number of states visited in a loop is 50, or 500 ms since the ECG data was sampled at 100 Hz. The model is not appropriate for heart rates above 120 bpm.
A special *outlier* state accommodates ECG-lead noise.

HMM Observation Model

Given that the system is in state s at time step t, and that the previous observations were $y[0], y[1], \ldots y[t-1]$ the model calculates a probabilistic forecast for the observation $y[t]$ as follows:

- The mean is an affine function of the past 3 observations:

$$\mu = a_0 + a_1 \cdot y[t-1] + a_2 \cdot y[t-2] + a_3 \cdot y[t-3]$$

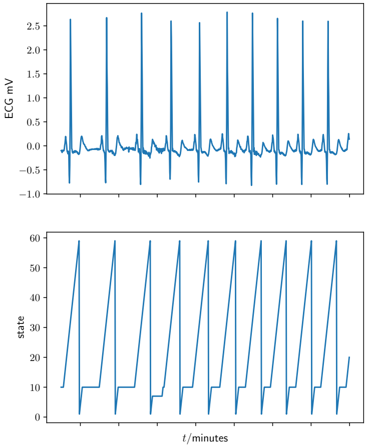
- The residual is Gaussian

$$y[t] \sim \mathcal{N}(\mu, \sigma^2)$$

- The parameters a and σ^2 are functions of the state s

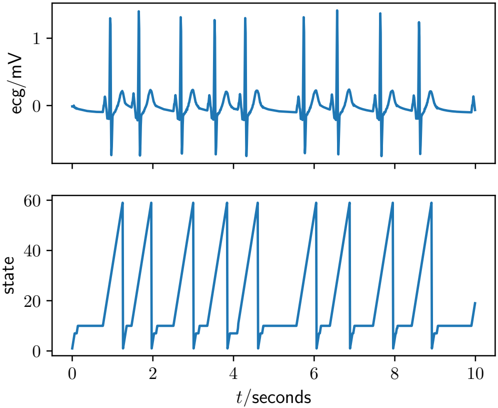
I used scipy.signal.find_peaks to supervise training of an initial model for one of the records from CINC 2000. I derived models for the other records from that initial model via unsupervised training.

Results



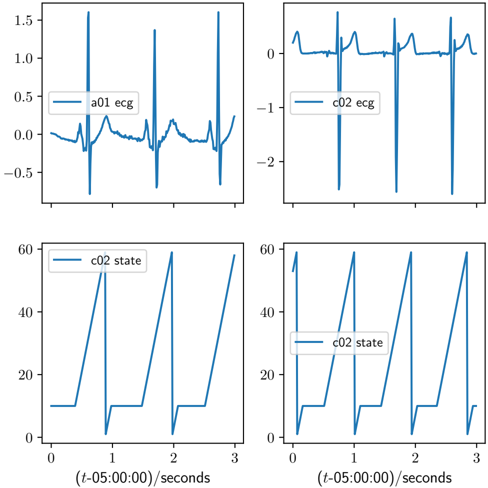
Variations in heart rate affect only the duration of residence in slow states.

Results



Driving models with a random number generator yields simulations that look plausible.

Results



The technique handles varying lead placement.

The End

Questions?