HMM Overview

Vitali Loseu*, Eric Guenterberg, Roozbeh Jafari
Embedded Systems and Signal Processing Lab
Department of Electrical Engineering, University of Texas at Dallas
Richardson, TX 75080-3021

*Note: Please direct your questions to vitali.loseu@gmail.com

- **Markov Process**

Markov process is a statistical process that describes a memory-less system, meaning that the probability of future events depends only on the current state, and not on any of the previous states. Markov processes often appear in the context of the exponential random distributions that also have memory-less property. This property can be described as the following conditional probability equality

$$Pr[X(t+\Delta t) = y \mid X(s) = x(s), \text{ for all } s \leq t] = Pr[X(t+\Delta t) = y \mid X(t) = x(t)]$$

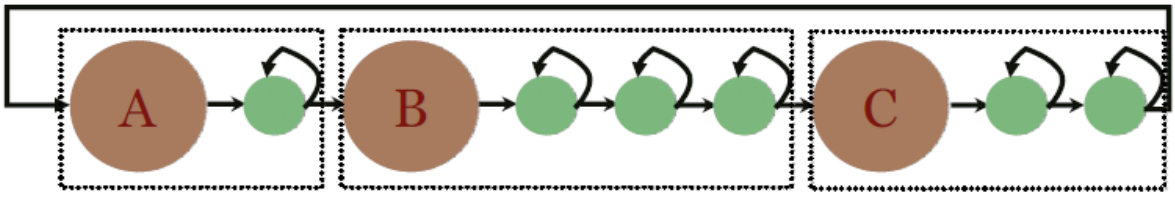
Which mean that that probability of being in state y at time $t+\Delta t$ conditioned on being in state x(t) at time t is equivalent to being in the same state y conditioned on state values before time t, which displays that any information before time t does not contribute to the probability.

In many cases, customer inter-arrival times exhibit memory-less property. This can be explained by the fact that there is potentially infinite number of customers, and each one of them determines their schedule themselves. Since the interval of the receiving system neither speed up nor slow down those independent arrivals the memory-less property must hold for the overall system.

- **Markov Chain**

In practice, Markov processes with a discreet state-space (also referred to as Markov chains), are used. Markov chain is a process that contains a discreet number of states, and describes transitions from state to state. According to the Markov memory-less property, only the current state and current observation affect the state transition.

There is a variety of properties and types of Markov chains. In this work, we are interested in a left-right Markov chain, where the model restricts what states can be accessed from different states. In our case states can only be accessed in the left to right direction, with a possible loop-around.



- **Hidden Markov Model**

In a normal Markov model, each state can be observed so that transition probabilities are the only parameters of the system. In the Hidden Markov Model states can not be directly observed, but the output of every state may be observed. Each state has probability distribution of outputs over a set of tokens. Therefore, based on the sequence of tokens generated by the outputs it is possible to evaluate a

particular state sequence with respect to others. In other words, the state sequence is also a parameter in addition to the transition probabilities between states.

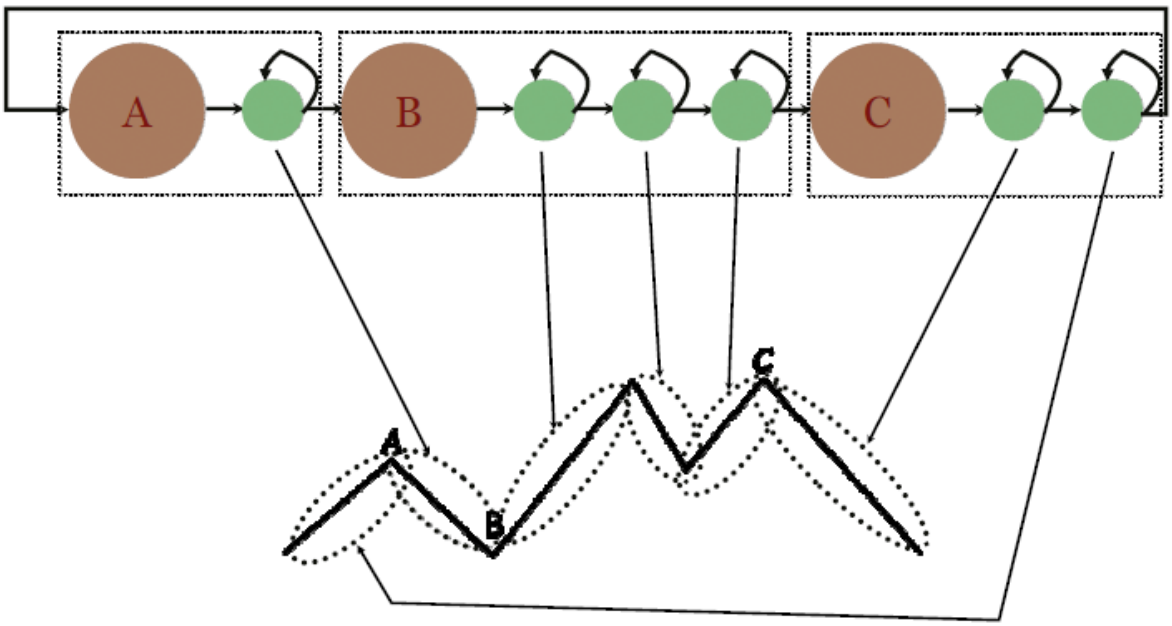
- **Hidden Markov Model For Annotation**

Based on the description of the Hidden Markov Model (HMM) multiple problems can be defined and addressed. We are interested in the problem of 1) finding the most likely state sequence based on a given output sequence or a set of such sequences. During the training part, the system finds the maximum likelihood estimate of parameters of the system given a known observation. 2) Given the parameters of the system and an output sequence, find the most likely state sequence. Based on the training data and the observed sequence, the system can annotate observed events by associating them with the states of the HMM. In here, the result of the first step is used for annotation, so it is essential to guarantee that the training of the system corresponds to the observation being annotated.

- **Hidden Markov Model Training**

During the training process, we collect observations, add manual annotation points, and define HMM parameters that correspond to those. This process is split in multiple stages:

- First, we define real states that correspond to events in the action, ie events we are trying to annotate, and transition states that define transition sequences between real states. Red states are real state, while green states are transition states. The system must transition from the real state after one observation, while it can self-transition to a transition state. The motivation of this design is displayed in the figure.



- All actions are divided into cohorts, where each cohort corresponds to one real state and a set of transition states to move to the next real state. Dotted boxes correspond to each cohort. Each cohort can be trained individually.

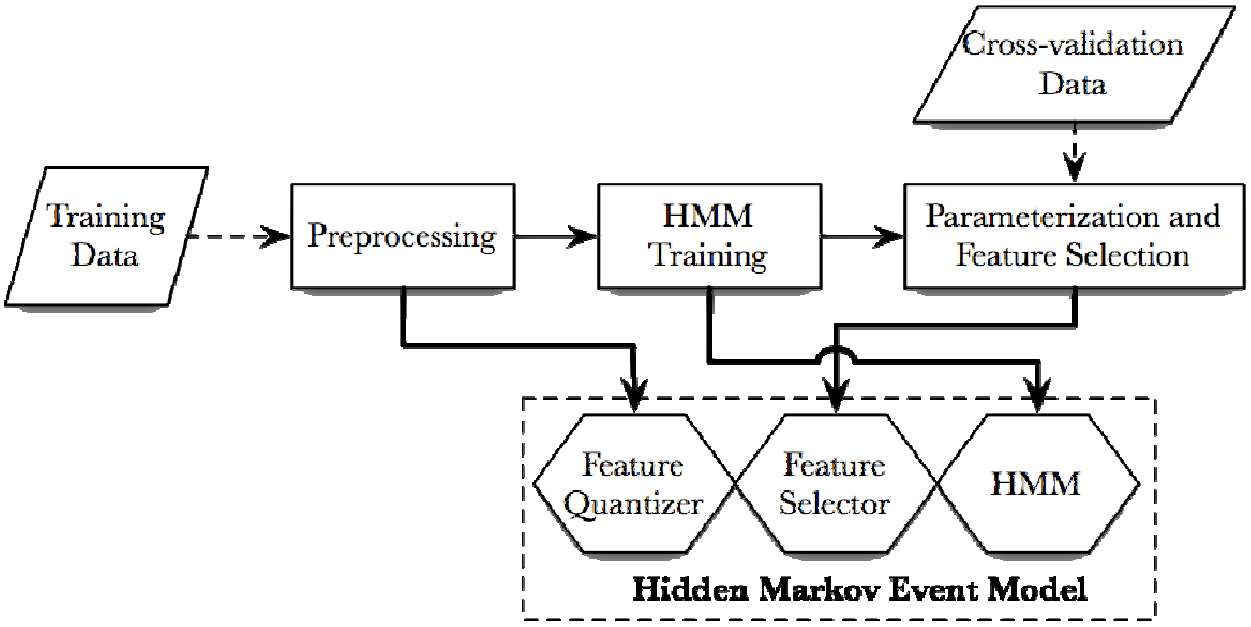
- Cohort training is defined with the following. First the system divides sample evenly among transition states and labels them accordingly. The labeled sequence is used to estimate the probabilities and update the HMM. The update model is then used to find the most likely state label sequence. The system re-estimates probabilities, updates HMM and repeats the process until termination condition is met.

- Once individual cohorts are trained, they can be combined to describe the overall model. This approach has a problem, since it is used to represent training data and not minimize error. It can be resolved by introducing parameters to cohorts. We introduced two parameters to each cohort:

1) Data features that describe the cohort.

2) Number of transition states.

Overall training process can be represented with



- **Hidden Markov Model Annotation**

After the system parameters are selected, the system is deployed to collect data and annotate it in real time. To do it we employed Viterby pathcounting algorithm. At any given time the system assesses probability of being in every possible state the system can be which includes both real and transition states. Viterby algorithm is used to select the best state sequence to represent the observation.