**ECE 8527: Introduction to
Machine Learning and Pattern Recognition**

# HW No. 7: Continuous Distribution Hidden Markov Models

For this assignment, we will use the package located here:

*https://github.com/jmschrei/pomegranate*

You might find this tutorial useful:

*https://pomegranate.readthedocs.io/en/latest/HiddenMarkovModel.html*

In this assignment, we will focus on modeling a sequence of random numbers – a 1D signal essentially. Hidden Markov Models (HMMs) model temporal relationships between events, allowing them to model the evolution of a signal’s spectrum as a function of time.

Construct a Python function to generate N random values of dimension $1$ that use a Gaussian mixture distribution with three mixture components. Use a random seed based on the time/date or the equivalent so that each time you call this function it generates a new sequence of random vectors.

Generate two sequences of 10,000 values using the following parameters:

Model 1: $μ\_{11}=\left[\begin{matrix}0.00\end{matrix}\right], C\_{11}=\left[\begin{matrix}1.00\end{matrix}\right]$, $μ\_{12}=\left[\begin{matrix}1.00\end{matrix}\right], C\_{12}=\left[\begin{matrix}1.00\end{matrix}\right]$, $μ\_{13}=\left[\begin{matrix}2.00\end{matrix}\right], C\_{13}=\left[\begin{matrix}1.00\end{matrix}\right],$
$$c\_{1}=\left[\begin{matrix}0.33&0.33&0.33\end{matrix}\right]$$

Model 2: $μ\_{21}=\left[\begin{matrix}1.00\end{matrix}\right], C\_{21}=\left[0.25\right]$, $μ\_{22}=\left[\begin{matrix}0.00\end{matrix}\right], C\_{22}=\left[\begin{matrix}0.5\end{matrix}\right]$, $μ\_{23}=\left[\begin{matrix}1.00\end{matrix}\right], C\_{23}=\left[\begin{matrix}0.75\end{matrix}\right],$
$$c\_{2}=\left[\begin{matrix}0.75&015.&0.10\end{matrix}\right]$$

Process both these scalars through a digital filter that consists of the following difference equation:

$$y\left[n\right]=0.75x\left[n-1\right]+0.25x\left[n-2\right]+x[n]$$

For those who don’t have a background in digital signal processing, this equation simply states that for each element of each input vector in your set, add the current value to scaled versions of the previous two values. Assume $x\left[n-1\right]$ and $x\left[n-2\right]$ are zero. The result of this operation will be two sequences of 10,000 vectors – set $1$ corresponds to data generated from model $1$, set $2$ from model $2$.

Train a three-state continuous distribution HMM with one Gaussian distribution per state (see the above pomegranate tutorial) on this data (train model $A$ on data set $1$, model $B$ on data set $2$). Initialize the models with random values. Perform enough iterations so that the model converges (which should happen in 10 iterations or less). Compute the likelihood each sequence was generated by each model, filling in this table:

|  |  |  |
| --- | --- | --- |
| **Model** | **Data Set** | **Log-Likelihood** |
| A | 1 | ????.???? |
| A | 2 | ????.???? |
| B | 1 | ????.???? |
| B | 2 | ????.???? |

Can you recognize which model (A or B) generated the data (set 1 or set 2)?

Next, we want to train HMM models in which we vary the number of states. Train fully ergodic HMM models starting with one state, two states, three states, etc. But to make this more interesting, generate a new set of 10,000 points from model $1$ and filter it. Call this set $3$. Also generate new filtered data from model $2$, which you can call set $4$. Since this data is different than the training data, but statistically similar, results should follow the previous table. Do not retrain the model on set $3$ or set $4$.

Complete the following table:

|  |  |
| --- | --- |
| **No. States** | **Log Likelihood** |
| **Train** | **Eval** |
| **M1 / Set 1** | **M2 / Set 2** | **M1 / Set 3** | **M1 / Set 4** | **M2 / Set 3** | **M2 / Set 4** |
| 1 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 2 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 3 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 4 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 5 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 6 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 7 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 8 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 9 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |
| 10 | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? | ????.???? |

Note that “M$1$ / Set $3$” means applying model $1$ trained on set 1 to data set $3$, and so forth. You then compute likelihoods that each data set was produced by each model. If the model fit is good, the likelihood will be high.

Explain what you learned about your model’s behavior from this exercise. How does the digital filter influence the choice of an “optimal” model?