NLP
Lab-3
Hidden Markov Model

Lab Objectives

- Introduce Hidden Markov Model
- Hidden Markov Model Applications
- Simple HMM example model

Hidden Markov Model

Hidden Markov Models are probability models that help programs come to the most likely decision, based on both previous decisions and current data.

Hidden Markov Model Applications

Hidden Markov Models are used in modern speech recognition systems. HMMs are also used in a bunch of other applications as well though:

- ✓ Speech synthesis (i.e., given a sentence, produce a series of sound bytes that likely correspond with that sentence)
- ✓ Part-of-speech tagging (e.g., in “She finished the race”, is “race” most likely a verb or a noun?)
- ✓ Machine translation
- ✓ Gene prediction

Simple example model

Suppose we want to determine the average annual temperature at a particular location on earth over a series of years. To make it interesting, suppose the years we are concerned with lie in the distant past, before thermometers were invented. Since we can’t go back in time, we instead look for indirect evidence of the

temperature. To simplify the problem, we only consider two annual temperatures, “hot” and “cold”.

Suppose that modern evidence indicates that the probability of a hot year followed by another hot year is 0.7 and the probability that a cold year is followed by another cold year is 0.6. We’ll assume that these probabilities held in the distant past as well. The information so far can be summarized as:

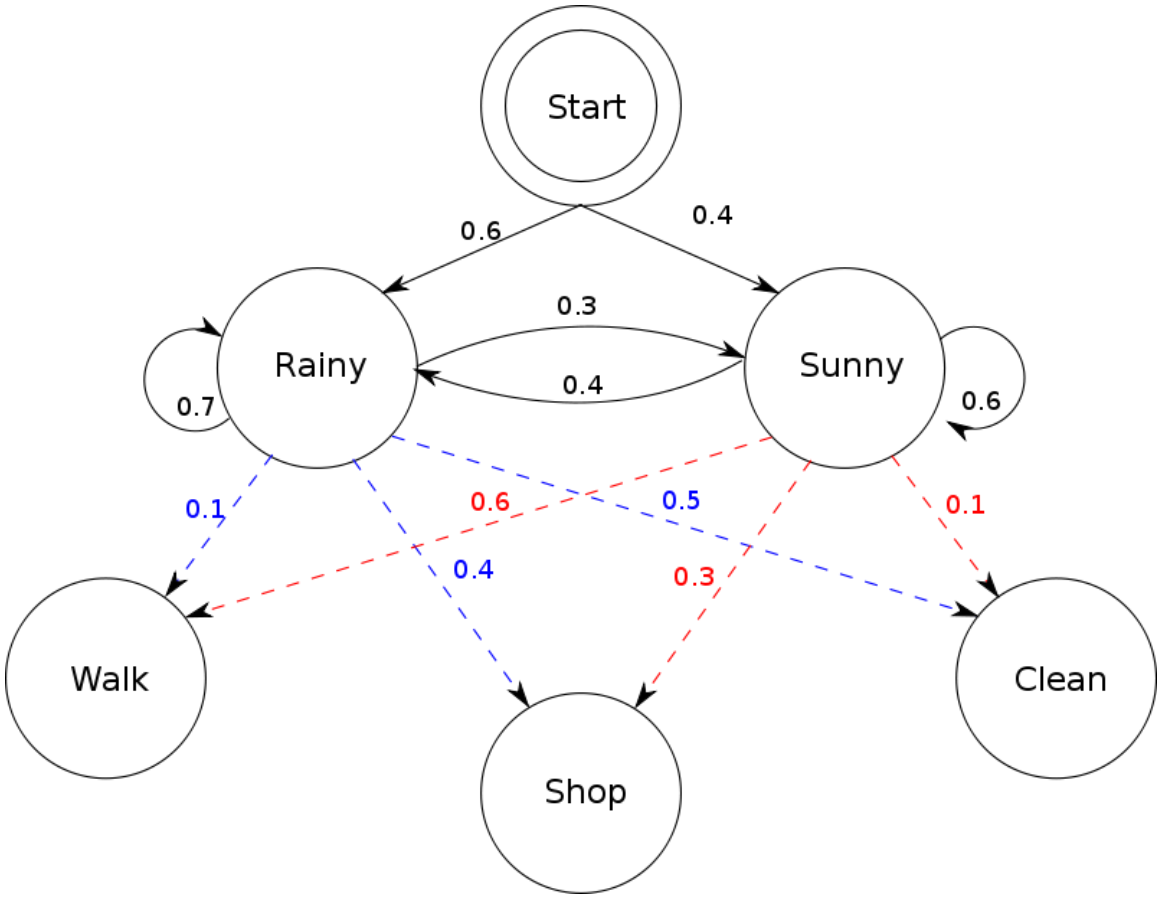
$$\begin{array}{c} \\ H \\ C \end{array} \begin{array}{c} \begin{array}{cc} H & C \end{array} \\ \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

Where H is “hot” and C is “cold”.

Also suppose that current research indicates a correlation between the size of tree growth rings and temperature. For simplicity, we only consider three different tree ring sizes, small, medium and large, or S, M and L, respectively. Finally, suppose that based on available evidence, the probabilistic relationship between annual temperature and tree ring sizes is given by:

$$\begin{array}{c} \\ H \\ C \end{array} \begin{array}{c} \begin{array}{ccc} S & M & L \end{array} \\ \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.4 & 0.5 \end{bmatrix} \end{array}.$$

The term hidden refers to the first order Markov process behind the observation. Observation refers to the data we know and can observe. Markov process is shown by the interaction between “Rainy” and “Sunny” in the below diagram and each of these are **HIDDE STATES**.



Observations: are known data and refers to “Walk”, “Shop”, and “Clean” in the above diagram. In machine learning sense, observation is our training data, and the number of hidden states is our hyper parameter for our model.

$$
\begin{aligned}
T &= \text{length of the observation sequence} \\
N &= \text{number of states in the model} \\
M &= \text{number of observation symbols} \\
Q &= \{q_0, q_1, \ldots, q_{N-1}\} = \text{distinct states of the Markov process} \\
V &= \{0, 1, \ldots, M-1\} = \text{set of possible observations} \\
A &= \text{state transition probabilities} \\
B &= \text{observation probability matrix} \\
\pi &= \text{initial state distribution} \\
\mathcal{O} &= (\mathcal{O}_0, \mathcal{O}_1, \ldots, \mathcal{O}_{T-1}) = \text{observation sequence.}
\end{aligned}
$$

T = don't have any observation yet, N = 2, M = 3,
Q = {"Rainy", "Sunny"},
V = {"Walk", "Shop", "Clean"}

The state transition matrix

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

comes from (1) and the observation matrix

$$B = \begin{bmatrix} \mathbf{0.6} & \mathbf{0.3} & \mathbf{0.1} \\ \mathbf{0.1} & \mathbf{0.4} & \mathbf{0.5} \end{bmatrix}$$

is from (2). In this example, suppose that the initial state distribution, denoted by π, is

$$\pi = \begin{bmatrix} 0.6 & 0.4 \end{bmatrix}.$$

State transition probabilities are the arrows pointing to each hidden state. **Observation probability matrix** is the blue and red arrows pointing to each observation from each hidden state. The matrix is row stochastic meaning the rows add up to 1.

How can we build the above model in Python?

```
states = ('Rainy', 'Sunny')

observations = ('walk', 'shop', 'clean')

start_probability = {'Rainy': 0.6, 'Sunny': 0.4}

transition_probability = {
   'Rainy' : {'Rainy': 0.7, 'Sunny': 0.3},
   'Sunny' : {'Rainy': 0.4, 'Sunny': 0.6},
   }

emission_probability = {
   'Rainy' : {'walk': 0.1, 'shop': 0.4, 'clean': 0.5},
   'Sunny' : {'walk': 0.6, 'shop': 0.3, 'clean': 0.1},
   }

from hmmlearn import hmm
import numpy as np

model = MultinomialHMM(n_components=2)
model.startprob_ = np.array([0.6, 0.4])
model.transmat_ = np.array([[0.7, 0.3],
                            [0.4, 0.6]])
model.emissionprob_ = np.array([[0.1, 0.4, 0.5],
                                [0.6, 0.3, 0.1]])
```

In the above case, emissions are discrete {“Walk”, “Shop”, “Clean”}. MultinomialHMM from the hmmlearn library is used for the above model. GaussianHMM and GMMHMM are other models in the library.

Now with the HMM what are some key problems to solve?

1. **Problem 1**: Given a known model what is the likelihood of sequence O happening?

```
import math

math.exp(model.score(np.array([[0]])))
# 0.30000000000000004
math.exp(model.score(np.array([[1]])))
# 0.36000000000000004
math.exp(model.score(np.array([[2]])))
# 0.3400000000000001
```

The probability of the first observation being “Walk” equals to the multiplication of the initial state distribution and emission probability matrix. 0.6 x 0.1 + 0.4 x 0.6 = 0.30 (30%). The log likelihood is provided from calling .score.

2. **Problem 2**: Given a known model and sequence O, what is the optimal hidden state sequence? This will be useful if we want to know if the weather is “Rainy” or “Sunny”

```
# predict a sequence of hidden states based on visible
states
logprob, seq = model.decode(np.array([[1,2,0]]).transpose())
print(math.exp(logprob))
print(seq)
logprob, seq = model.decode(np.array([[2,2,2]]).transpose())
print(math.exp(logprob))
print(seq)
```

```
0.01512
[0 0 1]
0.03675
[0 0 0]
```

Given the known model and the observation {“Shop”, “Clean”, “Walk”}, the weather was most likely {“Rainy”, “Rainy”, “Sunny”} with ~1.5% probability.

Given the known model and the observation {“Clean”, “Clean”, “Clean”}, the weather was most likely {“Rainy”, “Rainy”, “Rainy”} with ~3.6% probability.

Intuitively, when “Walk” occurs the weather will most likely not be “Rainy”.

References:

https://medium.com/@kangeugine/hidden-markov-model-7681c22f5b9

https://en.wikipedia.org/wiki/Hidden_Markov_model

https://nadesnotes.wordpress.com/2016/04/20/natural-language-processing-nlp-fundamentals-hidden-markov-models-hmms/