

Nama : Nur Hayati

NIM : G6601211015

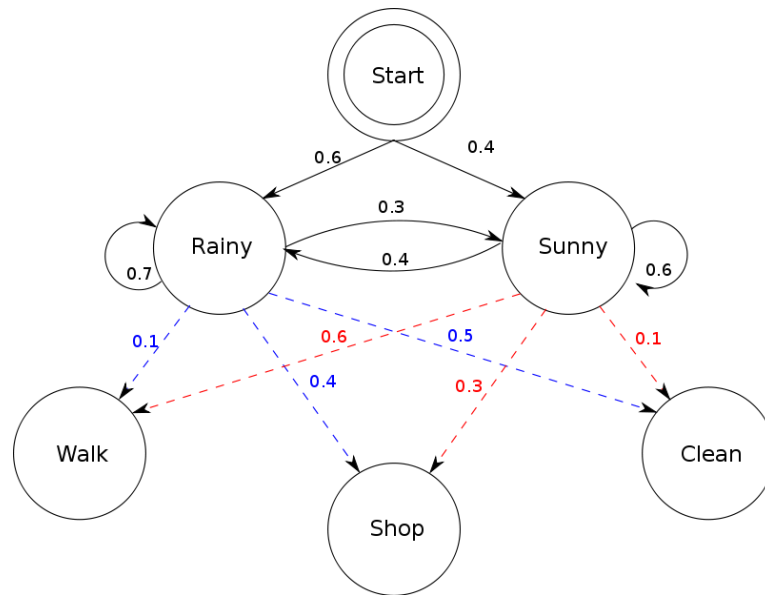
PERTEMUAN 5

MODEL HMM Menggunakan Python

TUJUAN PRAKTIKUM

Mahasiswa mampu mengimplementasikan model HMM dengan menggunakan python. HMM memiliki tiga permasalahan. Pada praktikum kali ini dua permasalahan akan dibahas yakni menentukan probability dari suatu observasi dan menentukan state optimal dari suatu observasi.

Adapun permasalahannya diilustrasikan pada diagram state sebagai berikut:



Gambar 1. Diagram State

Dari Gambar diperoleh beberapa informasi sebagai berikut:

1. Terdapat 2 hidden state : {Rainy, Sunny}
2. Terdapat 3 observable state : {Walk, Shop, Clean}
3. Terdapat nilai initial probability {0.6, 0.4}

Sebelum melakukan klasifikasi, terlebih dahulu kita melakukan instalasi library hidden markov (HMM) yaitu `hmmlearn`.

Cara melakukan instalasi library : `pip install hmmlearn==0.2.2`

1. Mendefinisikan State

```
# Definisi Matrik Transisi (states)

states = ('Rainy', 'Sunny')

# definisi Matriks Observasi / Matrik emmisi
observations = ('walk', 'shop', 'clean')

# definisi Matriks Priority
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},
}
```

2. Instalasi library dan representasi pada matrik

```
# Library untuk HMM
# Cara install pip install hmmlearn==0.2.2

from hmmlearn import hmm
import numpy as np

model = hmm.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]])
```

3. Menyelesaikan permasalahan 1 dari HMM

Given a known model what is the likelihood of sequence O happening?
The probability of the first observation being "Walk" equals to the multiplication

of the initial state distribution and emission probability matrix.
$0.6 \times 0.1 + 0.4 \times 0.6 = 0.30$ (30%).
The log likelihood is provided from calling .score.
catatan {'walk', 'shop', 'clean' }
Kasus untuk initial state
contoh untuk observasi 'walk'
peluang dari model jika observasi adalah 'walk'
$0.6 \times 0.1 + 0.4 \times 0.6 = 0.30$ (30%)

```
import math
math.exp(model.score(np.array([[0]])))
```

contoh untuk observasi 'shop'
peluang dari model jika observasi adalah 'shop'
$0.6 \times 0.4 + 0.4 \times 0.3 = (36\%)$

```
math.exp(model.score(np.array([[1]])))
```

Hasil :

```
0.30000000000000004 (walk)
0.36000000000000004 (shop)
```

contoh untuk observasi 'clean'
peluang dari model jika observasi adalah 'clean'
$0.6 \times 0.5 + 0.4 \times 0.1 = (34\%)$

```
math.exp(model.score(np.array([[2]])))
```

Hasil :

```
0.34000000000000001 (clean)
```

contoh untuk observasi {'clean', 'clean'}
peluang dari model jika observasi {'clean', 'clean'} adalah

```
math.exp(model.score(np.array([[2,2]])))
```

Hasil :

0.1244 (clean, clean)

4. Menyelesaikan permasalahan 2 dari HMM dengan viterbi algorithm

```
#==== HMM decode dengan algoritma viterbi di dalamnya =====#  
logprob, seq = model.decode(np.array([[1,2,0]]).transpose())  
print(math.exp(logprob))  
print(seq)
```

Hasil :

0.015120000000000003
[0 0 1]

**State terbaik untuk (shop, clean, walk) adalah saat
(rainy, rainy, sunny) dengan peluang sebesar 0.015 atau
1.5%**

```
logprob, seq = model.decode(np.array([[2,2,2]]).transpose())  
print(math.exp(logprob))  
print(seq)
```

Hasil :

0.036749999999999999
[0 0 0]

**State terbaik untuk (clean, clean, clean) adalah saat
(rainy, rainy, rainy) dengan peluang sebesar 0.0367 atau
3.67%**

TUGAS PRAKTIKUM

Latihan

1. Hitung peluang dari model jika observasi {'clean', 'clean', 'walk', 'walk', 'shop'} dari model yang ada
2. Tuliskan source code pada box di bawah ini

Hasil :

```
math.exp(model.score(np.array([[2,2,0,0,1]])))
```

Hasil : 0.0040823954000000001

1. Catat state yang paling baik untuk observasi {'clean', 'clean', 'walk', 'walk', 'shop'} dari model yang ada
2. Tuliskan source code pada box di bawah ini

Hasil :

```
logprob, seq =  
model.decode(np.array([[2,2,0,0,1]]).transpose())
```

```
print(math.exp(logprob))  
print(seq)
```

Hasil:

0.0012247200000000001

[0 0 1 1 1]

**State terbaik untuk (clean, clean, walk, walk, shop)
adalah saat (rainy, rainy, sunny, sunny, sunny) dengan
peluang sebesar 0.00122 atau 0.122%**

DAFTAR PUSTAKA

1. Richert W & Coelho LP. *Building Machine Learning System with Python*. 2013. Packt Publishing. Birmingham, UK.
2. <https://medium.com/@kangeugine/hidden-markov-model-7681c22f5b9>