Ex.no: 10

Design and Implementation of hidden Markov Models

Aim:

To Design and Implement the Hidden Markov Models.

Algorithm:

Step 1: We Multiply the Initial Probability of state i with the emission probability to observe o from state i at time t=1.

Step 2: we find the Maximum value among all the product results and Assign it to the viterbi variable.

Step 3: Generate Initial, Transition and Emission probability Distribution from the Sample Data.

Step 4: Generate, a List of all unknown Sequence.

Step 5: Score all unknown Sequences and select the best Sequence.

Step 6: Termination the following equation Depicts the probability of the Complete state sequence.

Code:-

```
import os
from hmm.hmm import DiscreteHMM
hidden_var_name = ('Sunny', 'foggy', 'rainy')
observation_var_name = ('no', 'yes')
hidden_var = {
```

```
hidden_var_name[0] : 0;
hidden_var_name[1] : 1;
hidden_var_name[2] : 2,
}
observation_var = {

        observation_var_name[0] : 0,
        observation_var_name[1] : 1,
    }
 target = [ ]
 obs_seq = [ ]

with open(os.path.join(os.path.dirname(__file__), 'input.txt')) as f:
 for line in f:
     hidden, observe = line.strip().split(',');
     target.append(hidden_var[hidden])
     obs_seq.append(observation_var[observe])
B = (
      (0.8, 0.2),
      (0.5, 0.5),
      (0.1, 0.9),
   )

hmm = DiscreteHMM(len(hidden_var), len(observation_var), B=B)

hmm.train(obs_seq, verbose = 1)
hmm.showmodel()
print('checksum:', hmm.check_model())
```

Result:

Thus the Design and Implementation of Hidden Markov Model is Succesfull and Verified the Output.

train.py - C:\Users\Smart\Downloads\weather_forecast\weather_forecast\train.py (3.9.7)

File Edit Format Run Options Window Help

```python
import os
from hmm.hmm import DiscreteHMM

# Mapping input to variable's id
hidden_var_name = ('sunny', 'foggy', 'rainy')
observation_var_name = ('no', 'yes')
hidden_var = {
    hidden_var_name[0]: 0,
    hidden_var_name[1]: 1,
    hidden_var_name[2]: 2,
    }

observation_var = {
    observation_var_name[0]: 0,
    observation_var_name[1]: 1,
    }

target = []
obs_seq = []

with open(os.path.join(os.path.dirname(__file__), 'input.txt')) as f:
    for line in f:
        hidden, observe = line.strip().split(',')
        target.append(hidden_var[hidden])
        obs_seq.append(observation_var[observe])

# Setting model
B = (
        (0.8, 0.2),
        (0.5, 0.5),
        (0.1, 0.9),
    )
hmm = DiscreteHMM(len(hidden_var), len(observation_var), B=B)

# Training the model best describe the observation
hmm.train(obs_seq, verbose=1)
hmm.show_model()
print('checksum:', hmm.check_model())
```

Ln: 1

```
========== RESTART: C:\Users\user\Desktop\ML\weather_forecast\train.py ==========
itnum     1 : delta 3.332041
itnum     2 : delta 0.428877
itnum     3 : delta 0.309993
itnum     4 : delta 0.215055
itnum     5 : delta 0.143531
itnum     6 : delta 0.094136
itnum     7 : delta 0.066443
itnum     8 : delta 0.052992
itnum     9 : delta 0.043964
itnum    10 : delta 0.037234
itnum    11 : delta 0.032206
itnum    12 : delta 0.028408
itnum    13 : delta 0.025485
itnum    14 : delta 0.023181
itnum    15 : delta 0.021314
itnum    16 : delta 0.019755
itnum    17 : delta 0.018418
itnum    18 : delta 0.017243
itnum    19 : delta 0.016188
itnum    20 : delta 0.015226
itnum    21 : delta 0.014339
itnum    22 : delta 0.013514
itnum    23 : delta 0.012741
itnum    24 : delta 0.012015
itnum    25 : delta 0.011332
itnum    26 : delta 0.010709
itnum    27 : delta 0.010213
itnum    28 : delta 0.009814
----------------------A: Transition probability----------------------
[[0.5434 0.4386 0.018 ]
 [0.5016 0.0854 0.413 ]
 [0.2865 0.1642 0.5494]]
----------------------B: Emission probability----------------------
[[0.9889 0.0111]
 [0.8347 0.1653]
 [0.1272 0.8728]]
-------------------pi: initital state distribution-------------------
[1. 0. 0.]
checksum: True
>>>
```