HW6

November 18, 2024

[1]:
```python
import numpy as np
import matplotlib.pyplot as plt
import string
```

[3]:
```python
pi = np.loadtxt("initialStateDistribution.txt", dtype=float)
a = np.loadtxt("transitionMatrix.txt", dtype=float)
b = np.loadtxt("emissionMatrix.txt", dtype=float)
o = np.loadtxt("observations.txt", dtype=int)
```

[5]:
```python
num_states = 27
T = len(o)
alphabet = dict(zip(range(1, 28), string.ascii_lowercase + ' '))
```

[7]:
```python
log_prob = np.zeros((num_states, T))
backtrack = np.zeros((num_states, T), dtype=int)
```

[9]:
```python
log_b = np.log(b)
log_a = np.log(a)
log_prob[:, 0] = np.log(pi[0]) + log_b[:, o[0]]
state_sequence = np.full(T, -1, dtype=int)
```

[11]:
```python
for t in range(1, T):
    prev_log_prob = log_prob[:, t - 1]
    for j in range(num_states):
        trans_probs = prev_log_prob + log_a[:, j]
        best_prev_state = np.argmax(trans_probs)
        log_prob[j, t] = trans_probs[best_prev_state] + log_b[j, o[t]]
        backtrack[j, t] = best_prev_state
```

[13]:
```python
state_sequence[T - 1] = np.argmax(log_prob[:, T - 1])
for t in range(T - 2, -1, -1):
    state_sequence[t] = backtrack[state_sequence[t + 1], t + 1]
```

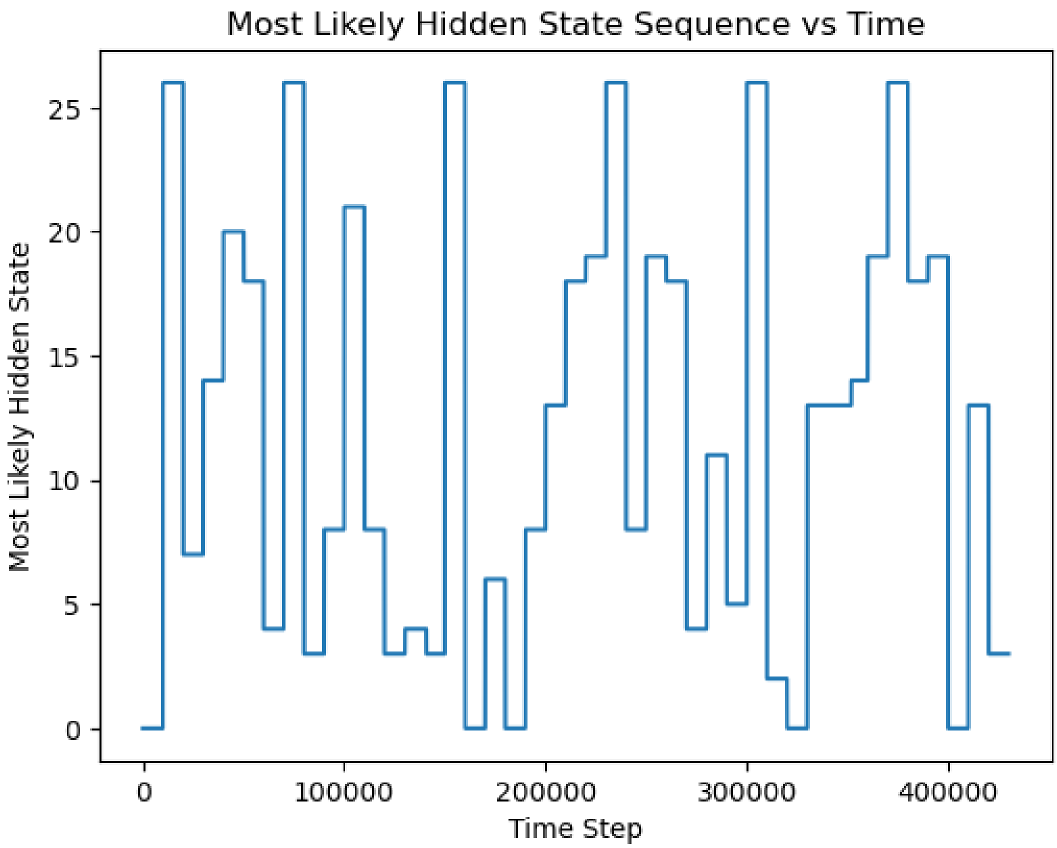
[15]:
```python
decoded_word = []
for t in range(T - 1):
    if state_sequence[t] != state_sequence[t + 1]:
        decoded_word.append(alphabet.get(int(state_sequence[t] + 1)))
decoded_word.append(alphabet.get(int(state_sequence[T - 1] + 1)))
```

```
decoded_word = ''.join(decoded_word)
```

[17]:
```
print(decoded_word)
```

a house divided against itself canot stand

[19]:
```
plt.plot(range(T), state_sequence)
plt.xlabel("Time Step")
plt.ylabel("Most Likely Hidden State")
plt.title("Most Likely Hidden State Sequence vs Time")
plt.show()
```



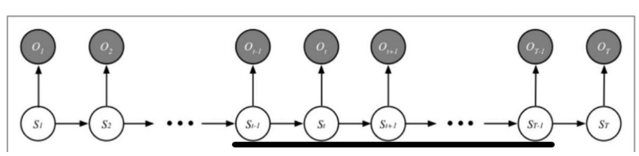
CSE 250A HW6

Akhil Nallacheruvu

November 2024

6.2 Conditional Independence

False	$P(S_t \mid S_{t-1})$	$=$	$P(S_t \mid S_{t-1}, S_{t+1})$
True	$P(S_t \mid S_{t-1})$	$=$	$P(S_t \mid S_{t-1}, O_{t-1})$
False	$P(S_t \mid S_{t-1})$	$=$	$P(S_t \mid S_{t-1}, O_t)$
False	$P(S_t \mid O_{t-1})$	$=$	$P(S_t \mid O_1, O_2, \ldots, O_{t-1})$
True	$P(O_t \mid S_{t-1})$	$=$	$P(O_t \mid S_{t-1}, O_{t-1})$
False	$P(O_t \mid O_{t-1})$	$=$	$P(O_t \mid O_1, O_2, \ldots, O_{t-1})$
True	$P(S_2, S_3, \ldots, S_T \mid S_1)$	$=$	$\prod_{t=2}^{T} P(S_t \mid S_{t-1})$
True	$P(S_1, S_2, \ldots, S_{T-1} \mid S_T)$	$=$	$\prod_{t=1}^{T-1} P(S_t \mid S_{t+1})$
False	$P(S_1, S_2, \ldots, S_T \mid O_1, O_2, \ldots, O_T)$	$=$	$\prod_{t=1}^{T} P(S_t \mid O_t)$
False	$P(S_1, S_2, \ldots, S_T, O_1, O_2, \ldots, O_T)$	$=$	$\prod_{t=1}^{T} P(S_t, O_t)$
True	$P(O_1, O_2, \ldots, O_T \mid S_1, S_2, \ldots, S_T)$	$=$	$\prod_{t=1}^{T} P(O_t \mid S_t)$
True	$P(O_1, O_2, \ldots, O_T)$	$=$	$\prod_{t=1}^{T} P(O_t \mid O_1, \ldots, O_{t-1})$



6.3 Belief Updating

a) $P(Y_1|X_1) = \Sigma_{X_0=x_0} P(Y_1|X_0 = x_0, X_1)P(X_0 = x_0)$

b) $P(Y_1) = \Sigma_{X_1=x_1} P(X_1 = x_1)\Sigma_{X_0=x_0} P(X_0 = x_0)P(Y_1|X_0 = x_0, X_1)$

c) $P(X_t|Y_1, ...Y_{t-1}) = P(X_t)$

$d) P(Y_t|X_t, Y_1, Y_2, ..., Y_{t-1}) = \Sigma_{X_{t-1}=x} P(Y_t, X_{t-1} = x|X_t, Y_1, ..., Y_{t-1}) =$
$\Sigma_{X_{t-1}=x} P(Y_t|X_{t-1} = x, X_t, Y_1, ..., Y_{t-1}) P(X_{t-1} = x|X_t, Y_1, ..., Y_{t-1}) =$
$\boxed{\Sigma_{X_{t-1}=x} P(Y_t|X_{t-1} = x, X_t) P(X_{t-1} = x|Y_1, ..., Y_{t-1})}$

$e) P(Y_t|Y_1, ..., Y_{t-1}) = \Sigma_{X_t=x_t} P(Y_t|X_t = x_t, Y_1, ..., Y_{t-1}) P(X_t = x_t) =$
$\boxed{\Sigma_{X_t=x_t} P(X_t = x_t) \Sigma_{X_{t-1}=x} P(Y_t|X_{t-1} = x, X_t) P(X_{t-1} = x|Y_1, ..., Y_{t-1})}$