Margot Wagner
A53279875
CSE 250a HW7
Due: 11/24/20

7.1 Viterbi Algorithm

In [62]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Viterbi algorithm

In [89]:
```
a = np.loadtxt('transitionMatrix.txt') # n x n
b = np.loadtxt('emissionMatrix.txt') # n x m
init = np.loadtxt('initialStateDistribution.txt') # n x 1 (pi)
n = len(init) # number of states
m = b.shape[1] # number of observations
o = np.loadtxt('observations.txt', dtype=int) # observations
T = len(o)
l = np.zeros((n,T))
phi = np.zeros_like(l)
```

In [90]:
```
print('a:', a.shape)
print('b:', b.shape)
print('init:', init.shape)
print('n:', n)
print('m:', m)
print('o:', o.shape)
print('T:', T)
print('l:', l.shape)
print('phi:', phi.shape)
```

```
a: (27, 27)
b: (27, 2)
init: (27,)
n: 27
m: 2
o: (175000,)
T: 175000
l: (27, 175000)
phi: (27, 175000)
```

In [91]:
```
def initialize_l():
    '''
    first step of filling in l* matrix
    '''
    l[:,0] = np.log(init) + np.log(b[:,o[0]])
```

In [92]:
```
def update_l(curr_t, next_t):
    '''
    fill l in from left to right given the current t timestep and the next timestep, t+1
    also creates theta for t+1
    '''
    next_l = np.max(np.add(l[:,curr_t], np.log(a)), axis=1) + np.log(b[:,o[next_t]])
    next_phi = np.asarray([np.argmax(l[:,curr_t] + np.log(a), axis=1)])

    return next_l, next_phi
```

In [93]:
```
# Initialize l* matrix
initialize_l()

# Fill l* matrix from left to right
for t in range(T-1):
    l[:,t+1], phi[:,t+1] = update_l(t, t+1)
```

In [95]:
```
s = np.zeros(T, dtype=int)
def initialize_s():
    s[-1] = np.argmax(l[:,-1])
```

In [96]:
```
def update_s(curr_t, next_t):
    '''
    computes most likely states by backtracking
    '''
    s_curr = phi[s[next_t], next_t]

    return s_curr
```

In [9]:
```python
# Initialize most likely states (s*) matrix
initialize_s()

# Fill s* matrix from right to left
for t in range(T-2, -1, -1):
    s[t] = update_s(t, t+1)
```

In [10]:
```python
def viterbi_letters(s):
    m = ""
    for c in s:
        if c == 26:
            m += " "
        else:
            m += chr(c + 97)
    return m
```

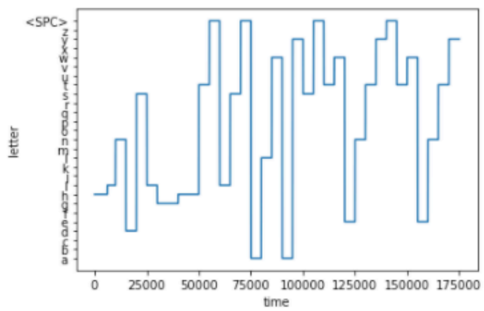
In [11]:
```python
letters = viterbi_letters(s)
```

In [12]:
```python
from itertools import groupby
message = ''
let_uniq = [i[0] for i in groupby(letters)]
for letter in let_uniq:
    message += letter
print(message)
```

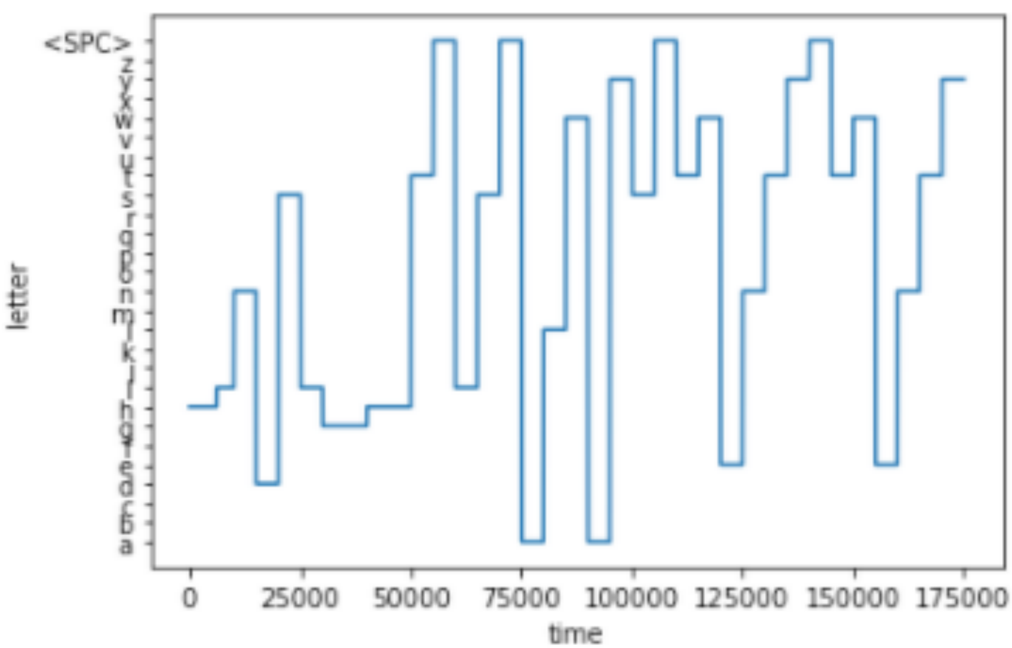
```
hindsight is always twenty twenty
```

In [13]:
```python
label = [chr(i) for i in range(97, 97+26, 1)]
label.append('<SPC>')
```

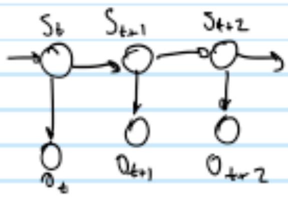
In [14]:
```python
plt.plot(s)
plt.xlabel('time')
plt.ylabel('letter')
plt.yticks(np.arange(0, 27, step=1), labels=label)
plt.show()
```



"hindsight is always twenty twenty"



7.2 Inference in HMM



$$\alpha_{it} = P(O_1 \sim O_t, S_t = i) \qquad a_{ij} = P(S_{t+1} = j \mid S_t = i)$$
$$\beta_{it} = P(O_{t+1}, O_{t+2}, \dots, O_T \mid S_t = i) \qquad b_{ik} = P(O_t = k \mid S_t = i)$$

(a) $P(S_t = i \mid S_{t+1} = j, O_1 \sim O_T)$

$$= \frac{P(S_t = i, S_{t+1} = j, O_1 \sim O_T)}{P(S_{t+1} = j, O_1 \sim O_T)}$$

$$P(S_{t+1} = j, O_1 \sim O_T) = P(O_1 \sim O_{t+1}, S_{t+1} = j) \cdot P(O_{t+2} \sim O_T \mid S_{t+1} = j)$$
$$= \alpha_{j,t+1} \cdot \beta_{j,t+1}$$

$$P(S_t = i, S_{t+1} = j, O_1 \dots O_T) = P(O_1 \dots O_t, S_t = i) \leftarrow \alpha_{it}$$
$$\cdot P(S_{t+1} = j \mid S_t = i, \cancel{O_1 \dots O_t}) \quad \text{cond ind} \rightarrow a_{ij}$$
$$\cdot P(O_{t+1} \mid S_{t+1} = j, \cancel{S_t = i, O_1 \dots O_t}) \quad \text{cond ind} \rightarrow b_j(O_{t+1})$$
$$\cdot P(O_{t+2} \dots O_T \mid S_{t+1} = j, \cancel{S_t = i, O_1 \dots O_{t+1}}) \quad \text{cond ind}$$
$$\hookrightarrow \beta_{j,t+1}$$

$$= \alpha_{it} a_{ij} b_{j,t+1} \beta_{j,t+1}$$

$\therefore$

$$P(S_t = i \mid S_{t+1} = j, O_1, \dots O_T) = \frac{\alpha_{it} a_{ij} b_{j,t+1} \cancel{\beta_{j,t+1}}}{\alpha_{j,t+1} \cancel{\beta_{j,t+1}}}$$

$$= \frac{\alpha_{it} a_{ij} b_j(O_{t+1})}{\alpha_{j,t+1}}$$

(b) $$P(S_{t+1} = j \mid S_t = i, O_1 \dots O_T) = \frac{P(S_t = i, S_{t+1} = j, O_1, \dots O_T)}{P(S_t = i, O_1 \dots O_T)}$$ product rule

$$P(S_t = i, O_1 \dots O_T) = P(S_t = i, O_1 \dots O_t) \cdot P(O_{t+1} \dots O_T \mid S_t = i)$$
$$= \alpha_{it} \beta_{it}$$

numerator equal to part (a) numerator

$$\Rightarrow \frac{\cancel{\alpha_{it}} a_{ij} b_{j,t+1} \beta_{j,t+1}}{\cancel{\alpha_{it}} \beta_{it}}$$

$$P(S_{t+1} = j \mid S_t = i, O_1 \dots O_T) = \frac{a_{ij} b_j(O_{t+1}) \beta_{j,t+1}}{\beta_{it}}$$

(c) $P(S_{t-1}=i, S_t=k, S_{t+1}=j \mid O_1, O_2 \dots O_T)$

$$= \frac{P(S_{t-1}=i, S_t=k, S_{t+1}=j, O_1, \dots O_T)}{P(O_1, \dots O_T)}$$ product rule

$P(O_1 \dots O_T) = \sum_\ell (O_1 \dots O_T, S_t=\ell)$ from part a

$= \sum_\ell \alpha_{\ell t} \beta_{\ell t}$

$P(S_{t-1}=i, S_t=k, S_{t+1}=j, O_1 \dots O_T) = P(O_1 \dots O_{t-1}, S_{t-1}=i)$

$\cdot P(S_t=k \mid \cancel{O_1 \dots O_{t-1}}, S_{t-1}=i)$ cond ind

$\cdot P(O_t \mid \cancel{O_1 \dots O_{t-1}, S_{t-1}=i}, S_t=k)$ cond ind

$\cdot P(S_{t+1}=j \mid \cancel{O_1 \dots O_t, S_{t-1}=i}, S_t=k)$ cond ind

$\cdot P(O_{t+1} \mid \cancel{O_1 \dots O_t, S_{t-1}=i, S_t=k}, S_{t+1}=j)$ cond ind

$\cdot P(O_{t+2} \dots O_T \mid \cancel{O_1 \dots O_{t+1}, S_{t-1}=i, S_t=k}, S_{t+1}=j)$ cond ind

$= P(O_1 \dots O_{t-1}, S_{t-1}=i) P(S_t=k \mid S_{t-1}=i) P(O_t \mid S_t=k) P(S_{t+1}=j \mid S_t=k)$

$\cdot P(O_{t+1} \mid S_{t+1}=j) P(O_{t+2} \dots O_T \mid S_{t+1}=j)$

$= \alpha_{i(t-1)} a_{ik} b_k(O_t) a_{kj} b_j(O_{t+1}) \beta_j(O_{t+1})$

$$P(S_{t-1}=i, S_t=k, S_{t+1}=j \mid O_1, O_2 \dots O_T) = \frac{\alpha_{i(t-1)} a_{ik} b_k(O_t) a_{kj} b_j(O_{t+1}) \beta_{j(t+1)}}{\sum_\ell \alpha_{\ell t} \beta_{\ell t}}$$

(d) $P(S_{t-1}=i \mid S_{t+1}=j, O_1 \dots O_T)$

$$= \frac{P(S_{t-1}=i, S_{t+1}=j, O_1 \dots O_T)}{P(S_{t+1}=j, O_1 \dots O_T)}$$ Product rule

$P(S_{t-1}=i, S_{t+1}=j, O_1 \dots O_T) = \sum_k P(S_{t-1}=i, S_t=k, S_{t+1}=j, O_1 \dots O_T)$ same from part c

$= \sum_k \alpha_{i(t-1)} a_{ik} b_k(O_t) a_{kj} b_j(O_{t+1}) \beta_j(O_{t+1})$

$= b_j(O_{t+1}) \beta_j(O_{t+1}) \alpha_{i(t-1)} \sum_k a_{ik} b_k(O_t) a_{kj}$

$P(S_{t+1}=j, O_1 \sim O_T) = P(O_1 \sim O_{t+1}, S_{t+1}=j) P(O_{t+2} \sim O_T \mid S_{t+1}=j)$

$= \alpha_{j,t+1} \beta_{j,t+1}$

$$P(S_{t-1}=i \mid S_{t+1}=j, O_1 \dots O_T) = \frac{\alpha_{i(t-1)} a_{ik} b_k(O_t) a_{kj} b_j(O_{t+1}) \cancel{\beta_j(O_{t+1})}}{\alpha_{j,t+1} \cancel{\beta_{j,t+1}}}$$

$$= \frac{b_j(O_{t+1}) \beta_j(O_{t+1}) \alpha_{i(t-1)} \sum_k a_{ik} b_k(O_t) a_{kj}}{\alpha_{j,t+1}}$$

7.3 Conditional Independence

$$\begin{tikzcd}
O_{t-1} & O_t & O_{t+1} \\
S_{t-1} \arrow[u] \arrow[r] & S_t \arrow[u] \arrow[r] & S_{t+1} \arrow[u]
\end{tikzcd}$$

$P(S_t \mid S_{t-1}) = P(S_t \mid S_{t-1}, O_t)$ FALSE

$P(S_t \mid S_{t-1}) = P(S_t \mid S_{t-1}, S_{t+1})$ FALSE

$P(S_t \mid S_{t-1}) = P(S_t \mid S_{t-1}, O_{t-1})$ TRUE

$P(S_t \mid O_{t-1}) = P(S_t \mid O_1, \dots O_{t-1})$ FALSE

$P(O_t \mid S_{t-1}) = P(O_t \mid S_{t-1}, O_{t-1})$ TRUE rule #2

$P(O_t \mid O_{t-1}) = P(O_t \mid O_1, O_2 \dots O_{t-1})$ FALSE

$P(O_1, O_2 \dots O_T) = \prod_{t=1}^{T} (O_t \mid O_1, \dots O_{t-1})$ TRUE product rule

$P(S_2, S_3 \dots S_T \mid S_1) = \prod_{t=2}^{T} P(S_t \mid S_{t-1})$ TRUE

$P(S_1, S_2 \dots S_{T-1} \mid S_T) = \prod_{t=1}^{T-1} P(S_t \mid S_{t+1})$ TRUE

$P(O_1, O_2 \dots O_T \mid S_1, S_2 \dots S_T) = \prod_{t=1}^{T} P(O_t \mid S_t)$ TRUE d-sep

$P(S_1, S_2 \dots S_T \mid O_1, O_2 \dots O_T) = \prod_{t=1}^{T} P(S_t \mid O_t)$ FALSE

$P(S_1, S_2 \dots S_T, O_1 \dots O_T) = \prod_{t=1}^{T} P(S_t, O_t)$ FALSE

7.4 Belief Updating

(a) $q_{jt} = p(S_t = j \mid O_1 \sim O_t)$ definition

$= p(S_t = j \mid O_1 \sim O_{t-1}, O_t)$

$= \dfrac{P(S_t = j \mid O_1 \sim O_{t-1}) \overbrace{P(O_t \mid S_t = j, \cancel{O_1 \sim O_{t-1}})}^{b_j(O_t)}}{P(O_t \mid O_1 \sim O_{t-1})}$ cond ind. Cond. Bayes rule

$P(S_t = j \mid O_1 \sim O_{t-1}) = \sum_i P(S_{t-1} = i, S_t = j \mid O_1 \sim O_{t-1})$ marginalization

$= \sum_i P(S_{t-1} = i \mid O_1 \ldots O_{t-1}) P(S_t = j \mid S_{t-1} = i, \cancel{O_1 \ldots O_{t-1}})$ cond ind product rule

$= \sum_i q_{it-1} a_{ij}$ subst definitions

$P(O_t \mid O_1 \ldots O_{t-1}) = \sum_j P(S_t = j, O_t \mid O_1 \ldots O_{t-1})$ marginalization

$= \sum_j \underbrace{P(O_t \mid S_t = j, O_1 \ldots O_{t-1})}_{b_j(O_t)} P(S_t = j \mid O_1 \ldots O_{t-1})$

$= \sum_{ij} b_j(O_t) a_{ij} q_{it-1} = Z_t$

$$\therefore \; q_{jt} = \frac{1}{Z_t} b_j(O_t) \sum_i a_{ij} q_{it-1}$$

(b)

$q_{jt} = \frac{1}{Z_t} b_j(O_t) \sum_i a_{ij} q_{it-1}$

$P(x_t \mid y_1, y_2 \ldots y_t) = \frac{1}{Z_t} P(y_t \mid x_t) \int$

$P(x_t \mid y_1 \ldots y_t) = P(x_t \mid y_1 \ldots y_{t-1}, y_t)$

$= \dfrac{P(x_t \mid y_1 \ldots y_{t-1}) P(y_t \mid x_t, \cancel{y_1 \ldots y_{t-1}})}{P(y_t \mid y_1 \ldots y_{t-1})}$ cond ind Bayes

$P(x_t \mid y_1 \ldots y_{t-1}) = \int dx_{t-1} P(x_t, x_{t-1} \mid y_1 \ldots y_{t-1})$ marginalization

$= \int dx_{t-1} P(x_{t-1} \mid y_1 \ldots y_{t-1}) P(x_t \mid x_{t-1}, \cancel{y_1 \ldots y_{t-1}})$ cond ind product rule

$= \int dx_{t-1} P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_1 \ldots y_{t-1})$

$$P(y_t \mid y_1 \dots y_{t-1}) = \int dx_t \, P(x_t, y_t \mid y_1 \dots y_{t-1})$$ marginalization

$$= \int dx_t \, P(y_t \mid x_t, \cancel{y_1 \dots y_{t-1}}) \, P(x_t \mid y_1 \dots y_{t-1})$$

conditional

$$= \int dx_t \, P(y_t \mid x_t) \int dx_{t-1} \, P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_1 \dots y_{t-1})$$

$$= Z_t$$

$\therefore$

$$P(x_t \mid y_1 \dots y_t) = \frac{1}{Z_t} P(y_t \mid x_t) \int dx_{t-1} \, P(x_t \mid x_{t-1}) P(x_{t-1} \mid y_1 \dots y_{t-1})$$

This is difficult because you have to integrate over the product of two probability distributions, which is not necessarily a closed form solution, but conditional probabilities of Gaussians are also Gaussian making it tractable.

7.5 V-Chain

(a) $P(Y_1 = j, O_1 = o_1) = \sum_i P(Y_1 = j, O_1 = o_1, X_1 = i)$ marginalization

$$= \sum_i P(Y_1 = j) P(X_1 = i \mid \cancel{Y_1 = j}) P(O_1 = o_1 \mid X_1 = i, Y_1 = j)$$

cond ind

$$= \sum_i \pi_j P(X_1 = i) b_{ij}(o_1)$$

(b) $\alpha_{j,t+1} = P(o_1 \ldots o_{t+1}, Y_{t+1} = j)$

$$= \sum_{k,i} P(Y_t = k, X_{t+1} = i, Y_{t+1} = j, o_1 \ldots o_{t+1})$$ marginalization

$$= \sum_{k,i} P(o_1 \ldots o_t, Y_t = k) P(X_{t+1} = i \mid Y_t = k, \cancel{o_1 \ldots o_t})$$ product rule, cond ind

$$\cdot P(Y_{t+1} = j \mid \cancel{Y_t = k, o_1 \ldots o_t, X_{t+1} = i})$$ cond ind

$$\cdot P(o_{t+1} \mid \cancel{Y_t = k, o_1 \ldots o_t}, X_{t+1} = i, Y_{t+1} = j)$$ cond ind

$$\alpha_{j,t+1} = \sum_{k,i} \alpha_{kt} a_{ki} \pi_j b_{ij}(o_{t+1})$$

(c) $P(o_1, o_2 \ldots o_T) = \sum_j (o_1 \ldots o_T, Y_T = j)$

$$= \sum_j \alpha_{jT}$$

(d) Likelihood for α_{jT} is a sum across Y's, n_y
Each α_{jT} is a sum across previous α's for each Y (n_y) and across all X's (n_x), so in total the complexity is

$O(n_y n_x)$ for a single step
$O(n_y n_x T)$ for all T steps