I am using 9 late hours on this set.

1a. First, we factorize $p(x \mid y = c)$:

$$p(x \mid y = c) = p(x_1, x_2, x_3, \dots, x_j \mid y = c)$$

$$= p(x_2, \dots, x_j \mid x_1, y = c)\, p(x_1 \mid y = c)$$

$$= p(x_3, \dots, x_j \mid x_1, x_2, y = c)\, p(x_2 \mid x_1, y = c)\, p(x_1 \mid y = c)$$

Repeat:

$$= \theta_{xjc}\theta_{x(j-1)c} \dots \theta_{x1c}$$

Since we stored each θ_{xjc}, we just need to account for the total stored per class. Since this grows with the binary nature of the input, we have the most complex input giving 2^{D-1} permutations. Thus, the big-O notation of this is $O(2^D)$, but we must consider this for each class. Thus, we need to consider $O(2^D * C)$ parameters.

1b. For this situation, we just have to consider the discrete x-y combinations. This means we just have to consider storing all values of x and y in order to calculate $p(x\,|y)$ for any combination. Thus, the input is a string of x's of length D. For each x, we can either have a 0 or 1, since it is binary. Therefore, we have 2^D possibilities for the input. Then, we have an output for each class. Our complexity is then $O(2^D * C)$, which is the same as part a.

1c. We would use the Naïve Bayes in this case. Since we are working with a very small sample size, there is a higher probability of having holes in the data set. The full model is a discriminative model, so it cannot tolerate missing values, which loses all guarantees. The Naïve Bayes model can deal with these missing features, so it would give a lower test set error.

1d. We would use the full model to give a lower test set error. The full model performs better than the Naïve version when there are no holes because the Naïve model uses conditional probability while the full model uses fully dependent features.

1e. For making a prediction using the Naïve model, we use

$$p(y \mid x) = \frac{p(x, y)}{p(x)} = \frac{p(x \mid y)p(y)}{p(x)}$$

We can compute $p(x \mid y)$ in $O(D)$ time since we know y and there are D values for x. We also know $p(y)$ is uniform, so it takes constant time to compute. Lastly, we have $p(x)$, which must be computed for each C. Thus, our overall complexity is $O(DC)$.

For making a prediction using the full model, we use

$$p(x \mid y) = \theta_{xjc}\theta_{x(j-1)c} \dots \theta_{x1c}$$

as computed in part a. Using the hint given in the problem, to compute the value we need for each C, we use the $O(D)$ operation that was described. We then apply this to each class, so our total computational complexity for prediction is $O(DC)$.

2a. My code generates:

```
################################################################################
                    Running Code For Question 2A
################################################################################


File #0:
Emission Sequence             Max Probability State Sequence
################################################################################
25421                         31033
01232367534                   22222100310
5452674261527433              1031003103222222
7226213164512267255           1310331000033100310
024712060235205101025524l     22222222222222222222222103


File #1:
Emission Sequence             Max Probability State Sequence
################################################################################
77550                         22222
7224523677                    2222221000
505767442426747               222100003310031
72134131645536112267          10310310000310333100
47336677714500510602530 41    22210000032222231032222 23


File #2:
Emission Sequence             Max Probability State Sequence
################################################################################
60622                         11111
4687981156                    2100202111
815833657775062               021011111111111
21310222515963505015          02020111111111111021
650319945257127400632002 5    11102021111111020211102 11


File #3:
Emission Sequence             Max Probability State Sequence
################################################################################
13661                         00021
2102213421                    3131310213
166066262165133               133333133133100
53164662112162634156          20000021313131002133
152354100512323022630625 6    13100213331331333133133 133
```

```
File #4:
Emission Sequence             Max Probability State Sequence
################################################################################
23664                         01124
3630535602                    0111201112
350201162150142               011244012441112
00214005402015146362          11201112412444011112
21112665246651435625344 50    20120124241240111124112411 24


File #5:
Emission Sequence             Max Probability State Sequence
################################################################################
68535                         10111
4546566636                    1111111111
638436858181213               110111010000011
13240338308444514688          00010000000111111100
01116644344413825336326 26    21111111111111100111110101
```

2b. Here are my results:

```
########################################################################
                    Running Code For Question 2Bi
########################################################################


File #0:
Emission Sequence            Probability of Emitting Sequence
########################################################################
25421                        4.537e-05
01232367534                  1.620e-11
5452674261527433             4.348e-15
7226213164512267255          4.739e-18
024712060235205101025524l    9.365e-24


File #1:
Emission Sequence            Probability of Emitting Sequence
########################################################################
77550                        1.181e-04
7224523677                   2.033e-09
505767442426747              2.477e-13
72134131645536112267         8.871e-20
473366777145005106025304l    3.740e-24


File #2:
Emission Sequence            Probability of Emitting Sequence
########################################################################
60622                        2.088e-05
4687981156                   5.181e-11
815833657775062              3.315e-15
21310222515963505015         5.126e-20
650319945257127400632002l    1.297e-25


File #3:
Emission Sequence            Probability of Emitting Sequence
########################################################################
13661                        1.732e-04
2102213421                   8.285e-09
166066262165133              1.642e-12
53164662112162634156         1.063e-16
152354100512323022630625l    4.535e-22

File #4:
Emission Sequence            Probability of Emitting Sequence
########################################################################
23664                        1.141e-04
3630535602                   4.326e-09
350201162150142              9.793e-14
00214005402015146362         4.740e-18
211126652466514356253445l    5.618e-22


File #5:
Emission Sequence            Probability of Emitting Sequence
########################################################################
68535                        1.322e-05
4546566636                   2.867e-09
638436858181213              4.323e-14
13240338308444514688         4.629e-18
011166443444138253363262l    1.440e-22
```

```
########################################################################
                    Running Code For Question 2Bii
########################################################################


File #0:
Emission Sequence            Probability of Emitting Sequence
########################################################################
25421                        4.537e-05
01232367534                  1.620e-11
5452674261527433             4.348e-15
7226213164512267255          4.739e-18
024712060235205101025524l    9.365e-24


File #1:
Emission Sequence            Probability of Emitting Sequence
########################################################################
77550                        1.181e-04
7224523677                   2.033e-09
505767442426747              2.477e-13
72134131645536112267         8.871e-20
473366777145005106025304l    3.740e-24


File #2:
Emission Sequence            Probability of Emitting Sequence
########################################################################
60622                        2.088e-05
4687981156                   5.181e-11
815833657775062              3.315e-15
21310222515963505015         5.126e-20
650319945257127400632002l    1.297e-25


File #3:
Emission Sequence            Probability of Emitting Sequence
########################################################################
13661                        1.732e-04
2102213421                   8.285e-09
166066262165133              1.642e-12
53164662112162634156         1.063e-16
152354100512323022630625l    4.535e-22

File #4:
Emission Sequence            Probability of Emitting Sequence
########################################################################
23664                        1.141e-04
3630535602                   4.326e-09
350201162150142              9.793e-14
00214005402015146362         4.740e-18
211126652466514356253445l    5.618e-22


File #5:
Emission Sequence            Probability of Emitting Sequence
########################################################################
68535                        1.322e-05
4546566636                   2.867e-09
638436858181213              4.323e-14
13240338308444514688         4.629e-18
011166443444138253363262l    1.440e-22
```

2c. My results:

```
Transition Matrix:
######################################################################
2.833e-01   4.714e-01   1.310e-01   1.143e-01
2.321e-01   3.810e-01   2.940e-01   9.284e-02
1.040e-01   9.760e-02   3.696e-01   4.288e-01
1.883e-01   9.903e-02   3.052e-01   4.075e-01


Observation Matrix:
######################################################################
1.486e-01   2.288e-01   1.533e-01   1.179e-01   4.717e-02   5.189e-02   2.830e-02   1.297e-01   9.198e-02   2.358e-03

1.062e-01   9.653e-03   1.931e-02   3.089e-02   1.699e-01   4.633e-02   1.409e-01   2.394e-01   1.371e-01   1.004e-01

1.194e-01   4.299e-02   6.529e-02   9.076e-02   1.768e-01   2.022e-01   4.618e-02   5.096e-02   7.803e-02   1.274e-01

1.694e-01   3.871e-02   1.468e-01   1.823e-01   4.839e-02   6.290e-02   9.032e-02   2.581e-02   2.161e-01   1.935e-02
```

2d. My results (I had to switch to Jupyter because numpy was not working for some reason):

```
######################################################################
                    Running Code For Question 2D
######################################################################


Transition Matrix:
######################################################################
5.413e-06   1.342e-01   8.658e-01   2.379e-08
1.269e-01   3.610e-01   2.221e-02   4.899e-01
3.634e-01   6.366e-01   4.555e-06   3.907e-09
3.501e-02   1.027e-04   3.197e-01   6.452e-01


Observation Matrix:
######################################################################
1.362e-01   7.629e-04   1.634e-01   1.769e-01   6.810e-03   3.249e-01   8.314e-03   3.654e-02   9.327e-02   5.301e-02
2.355e-01   1.144e-01   1.697e-01   3.305e-07   1.571e-01   6.108e-15   1.349e-01   3.375e-13   1.884e-01   2.590e-05
1.178e-01   6.175e-02   2.302e-41   1.560e-01   1.620e-01   1.034e-01   1.120e-01   1.037e-02   1.403e-01   1.363e-01
7.573e-02   6.812e-02   7.632e-02   1.293e-01   8.978e-02   7.933e-02   3.900e-02   2.643e-01   1.047e-01   7.342e-02
```

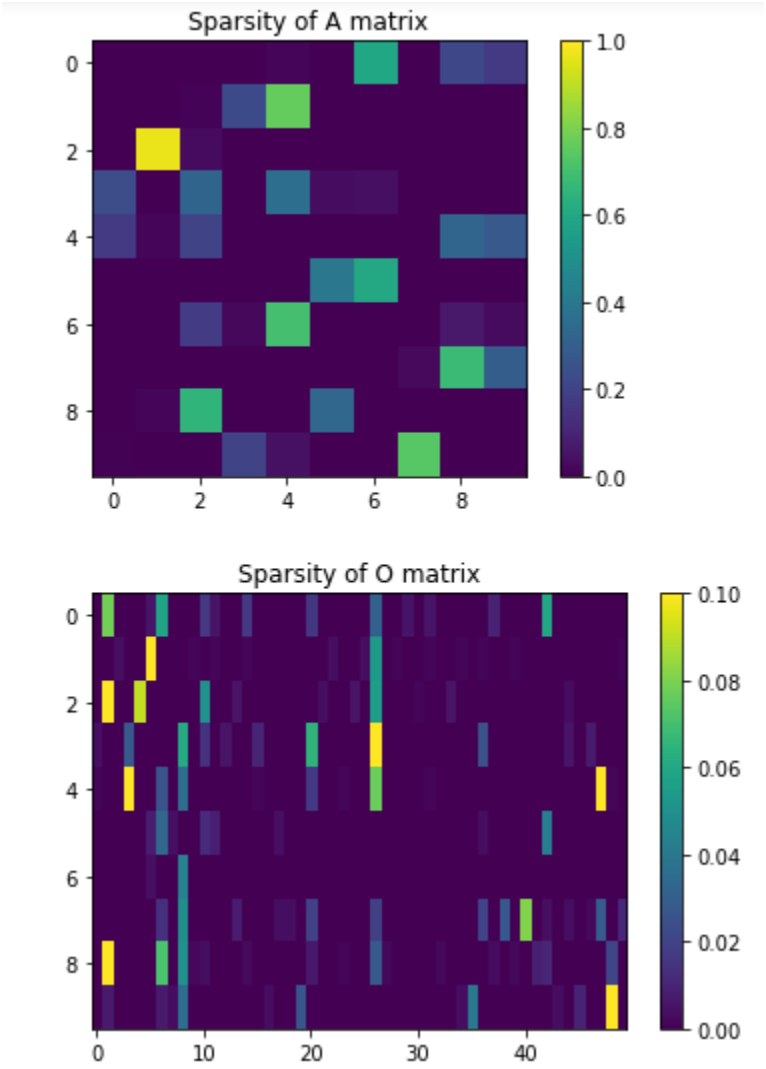
2e. It seems that the matrix from 2C provides a more accurate representation of Ron's mood and how they affect his music choices. First off, 2C is trained with fully supervised data while 2D is trained with unsupervised data, which is half of the data in 2C. When we are given the moods of the training data, it would make sense that we could better predict on the moods than if we did not. Also, in the transition matrix, the values on the principal diagonal are mostly larger in 2C than in 2D. It would make sense that a mood is more likely to stay in the current state given any situation. Lastly, the observation matrix for 2D has odd probabilities. The 2C observation matrix has values that are similar in degree, which makes sense in the context of preferring songs. The 2D matrix has some values to the power of negative 14, which seems off for the context of preferring songs.

We could improve the unsupervised method by increasing the training data set by getting more observations on Ron.

2f. My results:

```
##########################################################################
                    Running Code For Question 2F
##########################################################################


File #0:
Generated Emission
##########################################################################
55063173553563254152
74542515737124722425
74735573214332066454
14514405444424676755
40576534475554473746


File #1:
Generated Emission
##########################################################################
16411466245075770440
27166461771114312170
41512140473572456123
23245721614055032475
46752616661572554526


File #2:
Generated Emission
##########################################################################
93209130965216225262
81021349342967702774
21269827148451054220
23705234832197752546
30070661996600926231


File #3:
Generated Emission
##########################################################################
61650630442551120225
10251514514036156330
11051130004651026266
26136210142225261321
14212523362560263465

File #4:
Generated Emission
##########################################################################
36156034602343331510
66203500246551600515
66361655525611350342
16626062645253613042
50660662441013356626


File #5:
Generated Emission
##########################################################################
04081638644304824143
66448843643343888121
18316551416544663188
48688082568834465346
53044111151425461180
```

2g. The graphs generated:



The sparsity of both matrices is pretty obvious where the values are very sparse. There are a lot of zero values, and a majority of the nonzero values are very small. The sparsity in the A or transition matrix allows for easy transition. Since there are only a few values that are nonzero in each row, it shows the transitions that will be made at each state. This is the same for the O or observation matrix, there are only a few concrete options at each given state. Therefore, the large sparsity of each matrix allows for a few behaviors at each state.

2h. The samples emission sentences become more and more coherent as the number of hidden states increases. With a small number of hidden states, the words look almost random at each spot in the sentence. As we increase the number of hidden states, we start to see more phrases that would appear in the Constitution. For example, at four hidden layers, we get the phrase “the removal of the tax,” which makes sense in the order of nouns, verbs, and articles in sentence structure.

When there is only one hidden state, it seems that the words are chosen at random. Some of the words match adjacent words, but this seems out of luck rather than being trained. In general, we can increase the training data likelihood by allowing more hidden states, but this likelihood eventually cannot be increases when the number of hidden states gets large. The results of the visualization show that chunks of phrases become more likely with more hidden states, but the creation of these phrases will eventually plateau with the fixed observation set.

Some examples from my code:

```
In [4]: hmm1 = unsupervised_HMM(obs, 1, 100)
        print('\nSample Sentence:\n====================')
        print(sample_sentence(hmm1, obs_map, n_words=25))
```

```
Sample Sentence:
====================
The elections insure article acts chusing journal of any them be
the legislature court consist profit appointment convicted articl
e all to be exported of compact...
```

```
In [10]: hmm2 = unsupervised_HMM(obs, 2, 100)
         print('\nSample Sentence:\n====================')
         print(sample_sentence(hmm2, obs_map, n_words=25))
```

```
Sample Sentence:
====================
To and houses behaviour to states several into be such with and p
laces the law iv a as of with 4 miles supreme post them...
```

```
In [11]: hmm4 = unsupervised_HMM(obs, 4, 100)
         print('\nSample Sentence:\n====================')
         print(sample_sentence(hmm4, obs_map, n_words=25))
```

```
Sample Sentence:
====================
Shall temporary may when and nations proceedings of convicted mil
es their be of the removal of the tax of passed chuse to otherwis
e states vii...
```

2i. I will pick state four because of its large presence of verbs. This state represents a transition from some of the other states. Most of the other states have a lot of nouns like “treason” or “state,” so a lot of the states lack the transition between these nouns. State four has a lot of words like “holding,” “will,” and “may” that act as connecting words. If I were to guess, the transition matrix values from the noun states to this state would be higher with models with higher numbers of hidden states. This is shown in the final animation as there are bolded arrows from states that include words like “congress” and “people”