Machine Learning & Data Mining

Set 6 March 2023

2 Class-Conditional Densities for Binary Data [25 Points]

Problem A [5 points]:

View next page

A

We have $\mathbb{P}(x|y=c)$.

Using the chain rule, we have

$$\mathbb{P}(x|y=c) = \mathbb{P}(x_D | x_{1:D-1}, y=c) * \mathbb{P}(x_{D-1} | x_{1:D-2}, y=c) * \dots * \mathbb{P}(x_1 | y=c)$$

defining $\theta_{xjc} = \mathbb{P}(x_j | y=c)$ we'll have

$$\mathbb{P}(x|y=c) = \prod_{j=1}^{D} \theta_{xjc}$$

So, for each of the terms θ_{xjc} we will need to store a unique probability, for all the prefix with lenght $j-1$. Because of the data is binary, we will need to store 2^{j-1} parameters. This means that for each c, we will need to store $2^0 + 2^1 + 2^2 \dots + 2^{D-1}$ parameters. and thus, to represent this factorization we will need $O(C * 2^D)$ parameters.

Machine Learning & Data Mining

Set 6 March 2023

Problem B [5 points]:

For a give x with D binary features and C classes, it will be needed to store 2^D values. This means that it will be necessary to store $O(2^D * C)$ parameters in total for the probability $p(x \mid y = c)$ for arbitrary x and c. Comparing this with what we have in the previous item, we conclude that both of them have the same order of complexity.

For a given x with D binary features and C classes, there will be 2^D values to store. This means that we will need to store $O(2^D * C)$ parameters in order to being able to compute the conditional probability $p(x \mid y = c)$ for arbitrary x and c. Thus, comparing this with the answer from the previous part we have that both have the same complexity.

Problem C [2 points]:

If the sample size N is very small, it would be more difficult for the model to learn any relationship that can be between different features and this will cause the model doesn't generalize well. Considering that the assumption of Naïve Bayes is that features are independent, the Naïve Bayes model is likely to give a lower test error for the reason mentioned above. Also, having many parameters for a small training set can cause overfitting.

Problem D [2 points]:

Contrary to the previous case, for this scenario (large sample size) the full model is more likely to provide a lower test error. Considering that the Naïve model assume independency among features, it wont be able to learn the dependencies between features, which probably will lead to underfit (very simple model), while for the full model it will be able to generalize better when it has many parameters and will capture better the target probabilities.

Problem E [11 points]:

View next page

Ⓔ For Naive Bayes

$$P(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)} \Leftarrow \text{Baye's}$$

$$P(y|x) = \frac{\underbrace{p(x|y)}_{}^{\text{likelyhood}} \cdot \overset{\text{prior}}{p(y)}}{\sum_{c=1}^{C} p(x|y=c) \cdot p(y=c)} \Leftarrow \text{evidence}$$

- For $p(x|y)$ for a given x, requires multiplying D parameters $\rightarrow O(D)$

- Computing prior $P(y)$ is $O(1)$

- computing evidence $\sum_{c=1}^{C} p(x|y=c) \cdot p(y=c)$ requires to compute $p(x|y=c)$ for all $c \in \{1, 2, \ldots C\}$. $\therefore$ is $O(D*C)$

Thus, for naive Bayes we have complexity $O(D*C)$

For <u>Full model</u>

- Using again the Baye's Rule from previous part, we have that computing $p(x|y)$ requires converting the D-bit vector representation of x to an array index, which is an $O(D)$ operation according to the exercise.

Thus, this is an $O(D)$ operation (assuming a uniform class prior and converting a D-bit vector to an array index)

3 Sequence Prediction [75 Points]

https://colab.research.google.com/drive/1gykSYBYaHAtYphPaUtgtk3TKxbZiqBqM?usp=sharing

Problem A [10 points]:

File #0:

```
Emission Sequence          Max Probability State Sequence
######################################################################
25421                    31033
01232367534              22222100310
5452674261527433         1031003103222222
7226213164512267255      1310331000033100310
0247120602352051010255241    2222222222222222222222103
```

File #1:

```
Emission Sequence          Max Probability State Sequence
######################################################################
77550                    22222
7224523677               2222221000
505767442426747          222100003310031
72134131645536112267     10310310000310333100
4733667771450051060253041    2221000003222223103222223
```

File #2:

```
Emission Sequence          Max Probability State Sequence
######################################################################
60622                    11111
4687981156               2100202111
815833657775062          021011111111111
21310222515963505015     02020111111111111021
6503199452571274006320025    1110202111111102021110211
```

File #3:

```
Emission Sequence          Max Probability State Sequence
######################################################################
13661                    00021
```

```
2102213421              3131310213
166066262165133         133333133133100
53164662112162634156    20000021313131002133
1523541005123230226306256    1310021333133133133133133
```

File #4:

```
Emission Sequence        Max Probability State Sequence
######################################################################
23664                    01124
3630535602               0111201112
350201162150142          011244012441112
00214005402015146362     11201112412444011112
2111266524665143562534450    2012012424124011112411124
```

File #5:

```
Emission Sequence        Max Probability State Sequence
######################################################################
68535                    10111
4546566636               1111111111
638436858181213          110111010000011
13240338308444514688     00010000000111111100
0111664434441382533632626    2111111111111100111110101
```

Problem B [17 points]:

B.i Forward algorithm

```
File #0:

Emission Sequence          Probability of Emitting Sequence

#######################################################################

25421                   4.537e-05

01232367534             1.620e-11

5452674261527433        4.348e-15

7226213164512267255     4.739e-18

024712060235205101025524 1  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

4733667771450051060253041   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

6503199452571274006320025   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
1523541005123230226306256    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
2111266524665143562534450    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
0111664434441382533632626    1.440e-22
```

B.ii Backward algorithm

File #0:

```
Emission Sequence           Probability of Emitting Sequence
########################################################################
25421                  4.537e-05
01232367534               1.620e-11
5452674261527433           4.348e-15
7226213164512267255         4.739e-18
0247120602352051010255241    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
4733667771450051060253041    3.740e-24
```

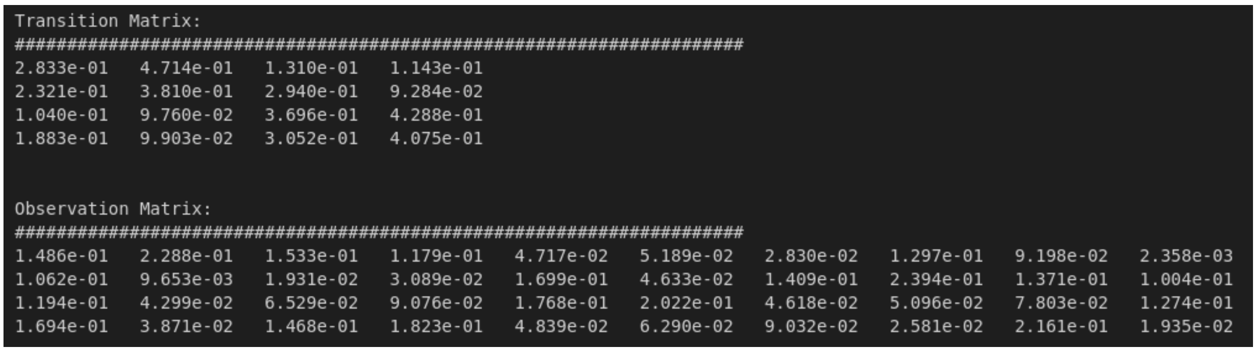
```
File #2:

Emission Sequence          Probability of Emitting Sequence

######################################################################

60622                2.088e-05

4687981156           5.181e-11

815833657775062      3.315e-15

21310222515963505015      5.126e-20

6503199452571274006320025    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

1523541005123230226306256    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

2111266524665143562534450    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

0111664434441382533632626    1.440e-22
```

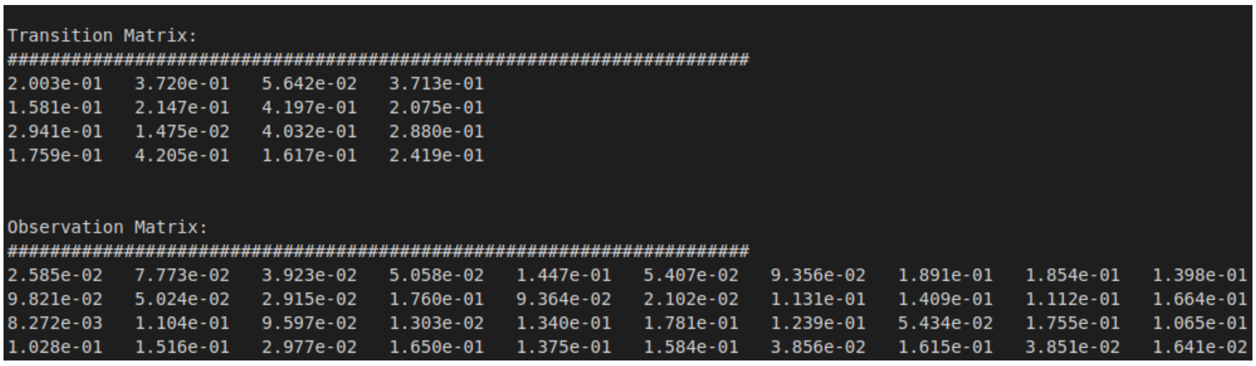
Problem C [10 points]:



```
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
```

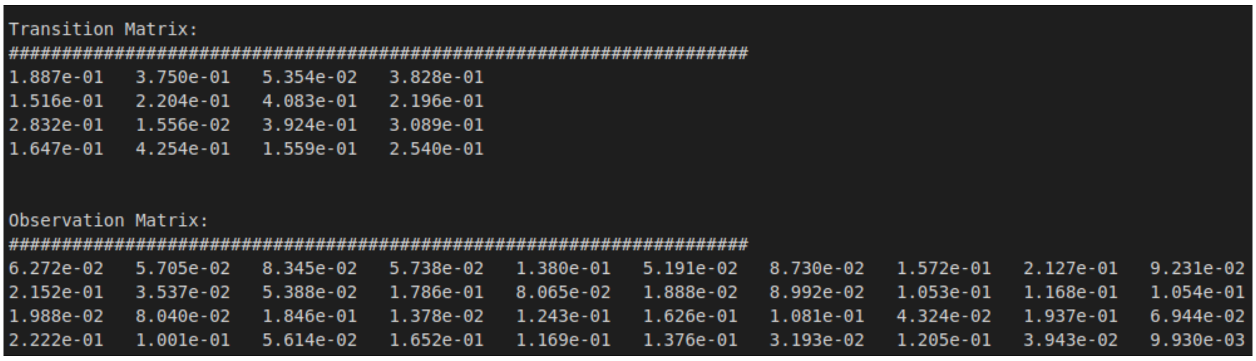
Problem D [15 points]:

0th iteration



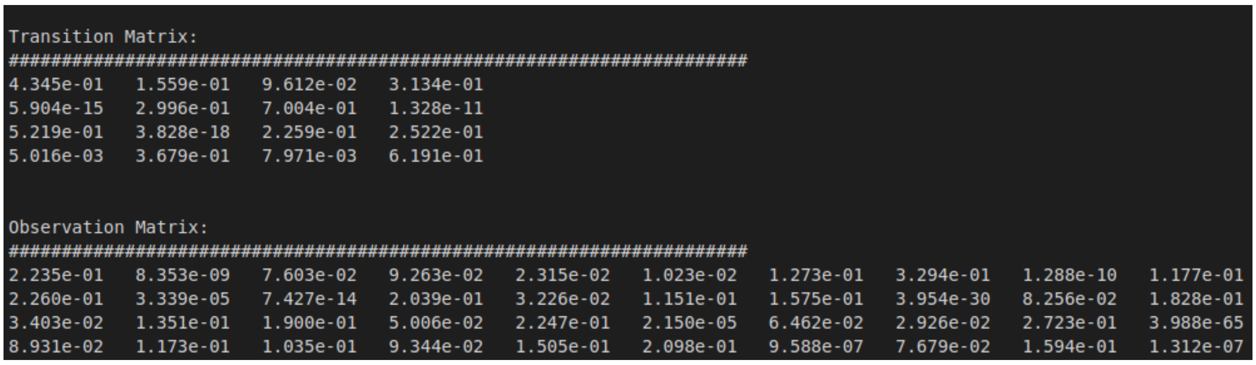
```
Transition Matrix:
######################################################################
2.003e-01   3.720e-01   5.642e-02   3.713e-01
1.581e-01   2.147e-01   4.197e-01   2.075e-01
2.941e-01   1.475e-02   4.032e-01   2.880e-01
1.759e-01   4.205e-01   1.617e-01   2.419e-01


Observation Matrix:
######################################################################
2.585e-02   7.773e-02   3.923e-02   5.058e-02   1.447e-01   5.407e-02   9.356e-02   1.891e-01   1.854e-01   1.398e-01
9.821e-02   5.024e-02   2.915e-02   1.760e-01   9.364e-02   2.102e-02   1.131e-01   1.409e-01   1.112e-01   1.664e-01
8.272e-03   1.104e-01   9.597e-02   1.303e-02   1.340e-01   1.781e-01   1.239e-01   5.434e-02   1.755e-01   1.065e-01
1.028e-01   1.516e-01   2.977e-02   1.650e-01   1.375e-01   1.584e-01   3.856e-02   1.615e-01   3.851e-02   1.641e-02
```

1st iteration



```
Transition Matrix:
######################################################################
1.887e-01   3.750e-01   5.354e-02   3.828e-01
1.516e-01   2.204e-01   4.083e-01   2.196e-01
2.832e-01   1.556e-02   3.924e-01   3.089e-01
1.647e-01   4.254e-01   1.559e-01   2.540e-01


Observation Matrix:
######################################################################
6.272e-02   5.705e-02   8.345e-02   5.738e-02   1.380e-01   5.191e-02   8.730e-02   1.572e-01   2.127e-01   9.231e-02
2.152e-01   3.537e-02   5.388e-02   1.786e-01   8.065e-02   1.888e-02   8.992e-02   1.053e-01   1.168e-01   1.054e-01
1.988e-02   8.040e-02   1.846e-01   1.378e-02   1.243e-01   1.626e-01   1.081e-01   4.324e-02   1.937e-01   6.944e-02
2.222e-01   1.001e-01   5.614e-02   1.652e-01   1.169e-01   1.376e-01   3.193e-02   1.205e-01   3.943e-02   9.930e-03
```

1000th iteration



```
Transition Matrix:
######################################################################
4.345e-01   1.559e-01   9.612e-02   3.134e-01
5.904e-15   2.996e-01   7.004e-01   1.328e-11
5.219e-01   3.828e-18   2.259e-01   2.522e-01
5.016e-03   3.679e-01   7.971e-03   6.191e-01


Observation Matrix:
######################################################################
2.235e-01   8.353e-09   7.603e-02   9.263e-02   2.315e-02   1.023e-02   1.273e-01   3.294e-01   1.288e-10   1.177e-01
2.260e-01   3.339e-05   7.427e-14   2.039e-01   3.226e-02   1.151e-01   1.575e-01   3.954e-30   8.256e-02   1.828e-01
3.403e-02   1.351e-01   1.900e-01   5.006e-02   2.247e-01   2.150e-05   6.462e-02   2.926e-02   2.723e-01   3.988e-65
8.931e-02   1.173e-01   1.035e-01   9.344e-02   1.505e-01   2.098e-01   9.588e-07   7.679e-02   1.594e-01   1.312e-07
```

Problem E [5 points]:

The A and O generated by the 1,000th iteration will be used to compare against A and O from part C. Upon observation, there are a few obvious differences.

The closed form solution for supervised learning allowed us to produce matrices from part C much faster than those from part D (0.01 vs 35.6 seconds).

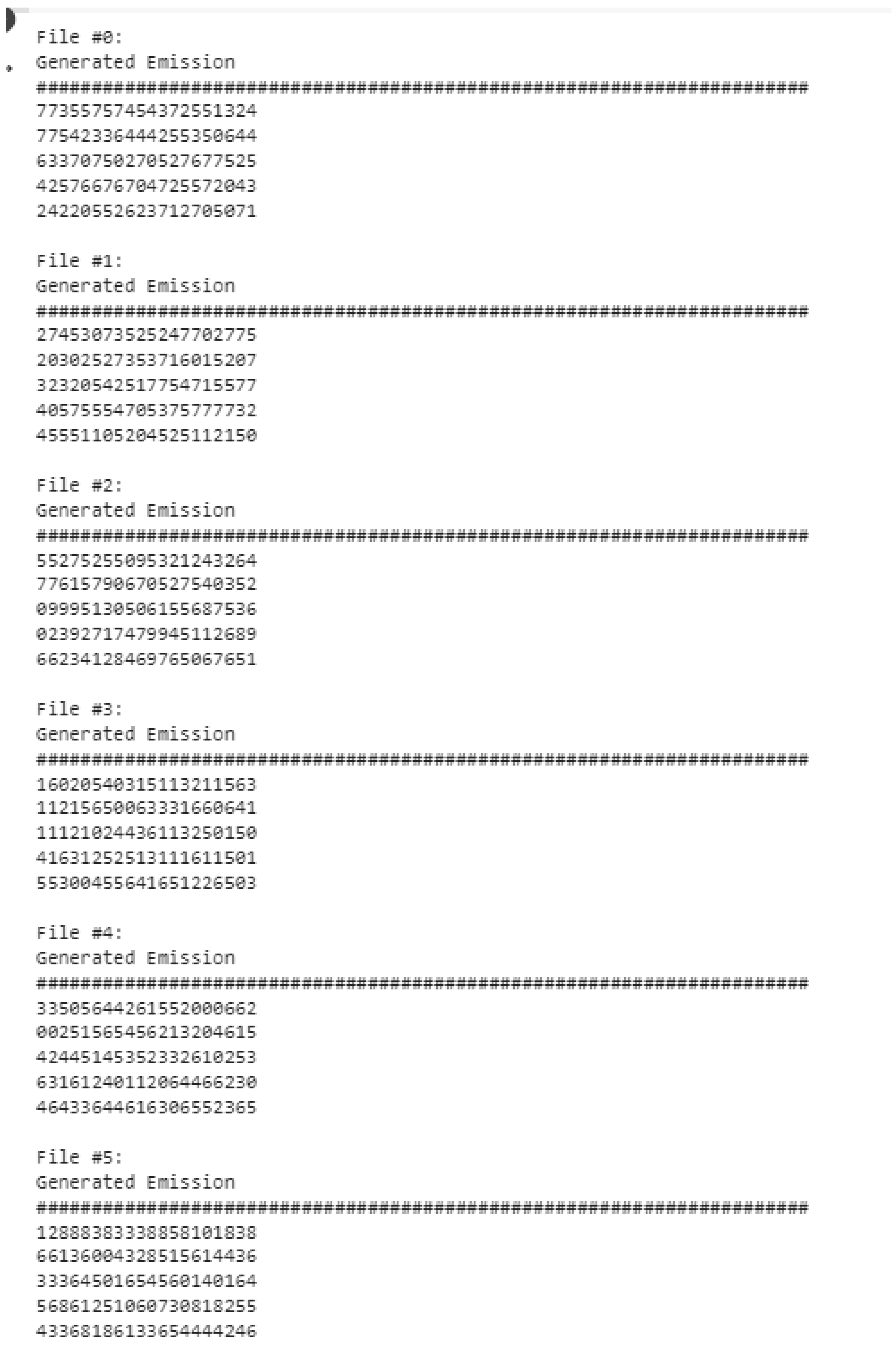
We see that many of the values in part D matrices actually converged to zero, whereas the values in matrices in part C are only two to three significant figures away. Thus, we can see that part D matrices are much more sparse than matrices from part C. We develop a notion of sparsity for part D matrices.

From class, we saw the closed form solution of supervised learning is the global optimal solution. This produces the same solution. The nonconvexity of Baum-Welch initialized at random spots in the A and O space may produce convergence to local optima. This produces different solutions.

Since we have labeled data in part C, and we have a certificate of global optima, the matrices from part C should be a more accurate representation of Ron's moods. The HMM states in the labeled data are directly Ron's moods. Conversely, we don't know what the hidden states from part D represent.

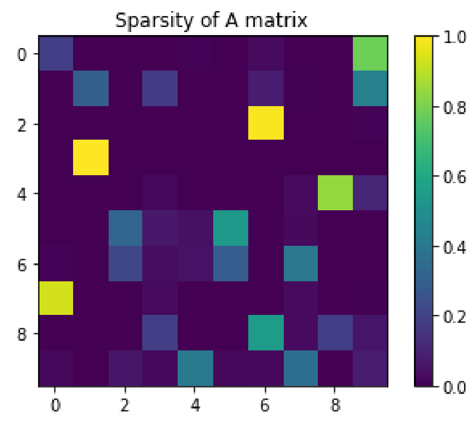
A potential improvement for unsupervised learning is to determine semantic meaning in the HMM states. It goes without saying that this is the major pitfall in the unsupervised learning framework. Perhaps a mathematical representation to describe semantic meaning of the hidden states could improve the algorithm.

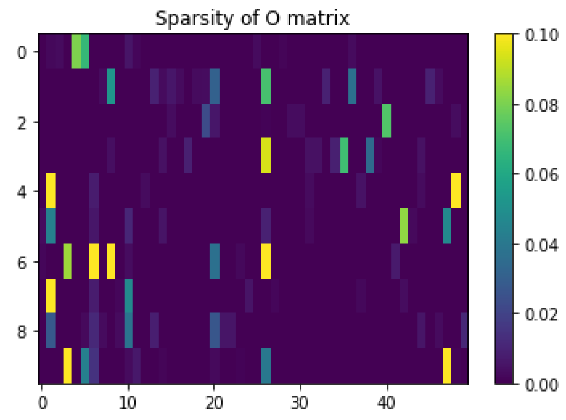
Problem F [5 points]:



```
File #0:
Generated Emission
######################################################################
77355757454372551324
77542336444255350644
63370750270527677525
42576676704725572043
24220552623712705071

File #1:
Generated Emission
######################################################################
27453073525247702775
20302527353716015207
32320542517754715577
40575554705375777732
45551105204525112150

File #2:
Generated Emission
######################################################################
55275255095321243264
77615790670527540352
09995130506155687536
02392717479945112689
66234128469765067651

File #3:
Generated Emission
######################################################################
16020540315113211563
11215650063331660641
11121024436113250150
41631252513111611501
55300455641651226503

File #4:
Generated Emission
######################################################################
33505644261552000662
00251565456213204615
42445145352332610253
63161240112064466230
46433644616306552365

File #5:
Generated Emission
######################################################################
12888383338858101838
66136004328515614436
33364501654560140164
56861251060730818255
43368186133654444246
```

Problem G [3 points]:



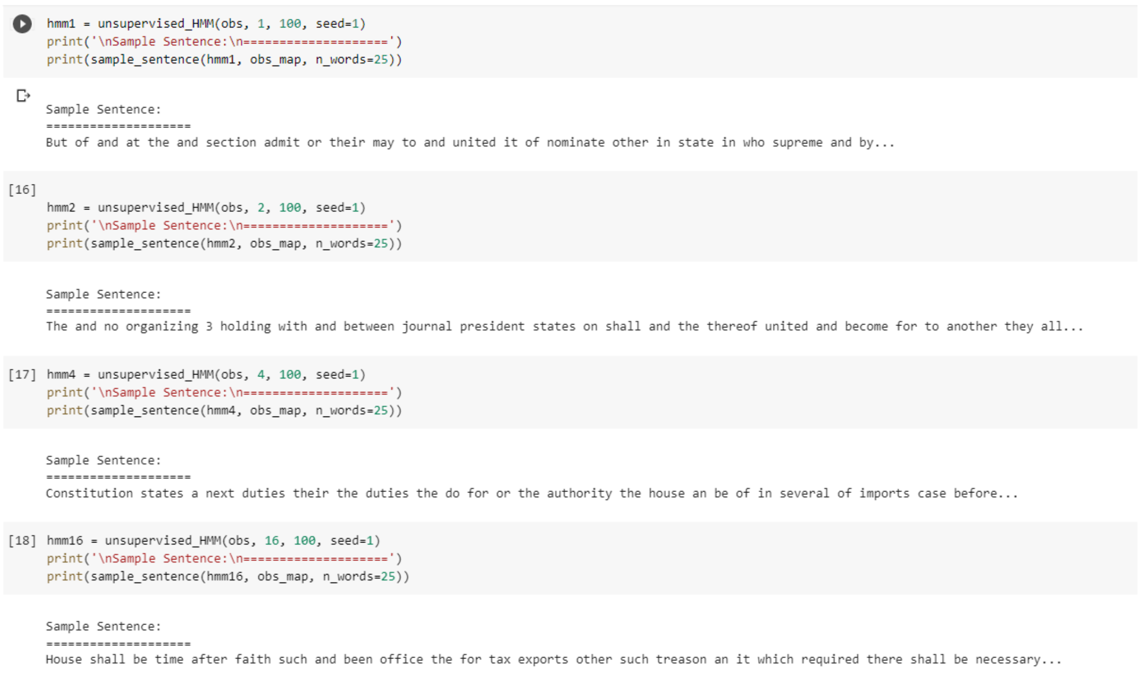


The visualizations show that the A and O matrices are extremely sparse. With see that there is significant sparsity in the diagonal of A. The O matrix also shows consistent sparsity but along some of the columns.

For the transition matrix, A, an element A[i,j] represents the probability of state i to go to state j. From our previous observation, the sparsity visualization tells us that it is very unlikely for state i to come back to itself. Hence, the sparsity in the diagonal of A. We see that there is a high likelihood of lower states going to higher states (lower and higher meaning low i integer and high j integer). This can be seen in the upper right triangular matrix of A. The same can be said about the lower triangular matrix of A.

For the observation matrix, O, an element O[i,j], represents that a state i is likely to produce the observation j. As previously mentioned, we see sparsity along columns. Therefore, some observations are very unlikely to be produced by all states. This corresponds to the sparsity of columns in O. For example, columns j = [10,20] and j = [30,40] are very unlikely to be generated by all states.

Problem H [5 points]:



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

```
Sample Sentence:
====================
But of and at the and section admit or their may to and united it of nominate other in state in who supreme and by...
```

```
[16]
hmm2 = unsupervised_HMM(obs, 2, 100, seed=1)
print('\nSample Sentence:\n====================')
print(sample_sentence(hmm2, obs_map, n_words=25))
```

```
Sample Sentence:
====================
The and no organizing 3 holding with and between journal president states on shall and the thereof united and become for to another they all...
```

```
[17] hmm4 = unsupervised_HMM(obs, 4, 100, seed=1)
print('\nSample Sentence:\n====================')
print(sample_sentence(hmm4, obs_map, n_words=25))
```

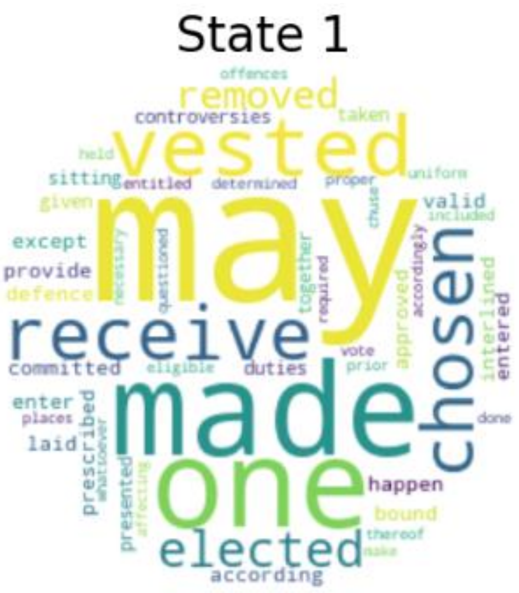
```
Sample Sentence:
====================
Constitution states a next duties their the duties the do for or the authority the house an be of in several of imports case before...
```

```
[18] hmm16 = unsupervised_HMM(obs, 16, 100, seed=1)
print('\nSample Sentence:\n====================')
print(sample_sentence(hmm16, obs_map, n_words=25))
```

```
Sample Sentence:
====================
House shall be time after faith such and been office the for tax exports other such treason an it which required there shall be necessary...
```

Overall, the sentences produced are not fully intelligible. However, as expected, we see that the sentences make more sense as the number of hidden states increases. With one hidden state, it almost seems like a random soup of words. Higher number of hidden states produce sentences that resemble sentences from the Constitution. We can increase the training data likelihood by allowing more hidden states since we can make stronger predictions in transitions in states and words.

Problem I [5 points]:



State 1 produced interesting results. The word cloud includes words like: *vested, chosen, receive, made, elected* which are heavily related by a notion of being assigned to a position. As the United States is based on a democracy, this state accurately captured an important aspect of a democracy–representation of the people via elected officials.

We see that some states describe something qualitative such as a states that seems to describe *unity* (keywords: “united”, “state”, “congress”). Other states describe things quantitatively such as states that describe *numbers* (keywords: “one”, “two”, “thirty”). Other states that were produced appear to not form a semantic meaning. Looks like they may have formed based on the frequency of the words.