* 2 late hours used

CS 155 PS 6

1 Class-Conditional Densities for Binary Data

Problem A

$p(x \mid y = c)$ can be factorized by chain rule of probability as

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

$= \theta_{xDc} * \theta_{x(D-1)c} * \dots * \theta_{x1c}$ (define θ_{x1c} as $P(x_1 \mid y = c)$ for simplicity)

$= \Pi_{j=1}^{D} \theta_{xjc}$

Storing each θ_{xjc} requires storing a unique probability for every possible prefix of length j-1, thus requiring 2^{j-1} parameters (since features are binary). So storing all the θ_{xjc} for j from 1 to D requires $2^0 + 2^1 + \dots + 2^{D-1}$ parameters for each c, and to store such values for all C classes, we need $O(C * 2^D)$ parameters.

Problem B

There are 2^D possible values for x with D binary features, and C classes. This requires storing O(C * 2^D) parameters to be able to compute p(x | y = c) for arbitrary x and c, which is the same complexity as the one from part A.

Problem C

The Naive Bayes model is likely to give lower test set error when the sample size small, since having too many parameters while fitting a small training set could cause overfitting. When there is only a small number of training inputs, trying to learn any relationship between different features in an input most likely would not generalize very well, and would be better to assume conditional independence among all the features.

Problem D

When the sample size N is very large, the full model is likely to give lower test set error. Assuming that the sample is a fairly accurate representation of the population and that the features are dependent on each other, using many parameters would allow the model to capture the target probabilities better, while the Naive model is unable to learn the feature dependencies and will underfit.

Problem E

Naive Bayes: O(D * C)

We know that p(y | x) = p(x | y) p(y) / p(x) by Bayes' Rule, and p(x) = $\Sigma_{c=1}^{C}$ p(x | y = c) p(y = c). Computing p(x | y) for a given x, y requires multiplying D parameters (O(D)), computing a uniform class prior p(y) is O(1), and computing p(x) = $\Sigma_{c=1}^{C}$ p(x | y = c) p(y = c) requires computing p(x | y = c) for all c ∈ {1, 2, …, C}, which is O(D * C).

Full model: O(D + C)

Once again, use Bayes' Rule to compute p(y | x) for a given x, y, using p(x | y = c) computed for arbitrary x and c and stored (i.e. a matrix with 2^D rows and C columns). Computing p(x | y) requires converting the D-bit vector representation of x to an array index (O(D)) and looking up the value for the corresponding row number (obtained array index) and column number (value of y) in the stored matrix (O(1)). Computing p(y) is O(1) assuming uniform probability, and computing p(x) = $\Sigma_{c=1}^{C}$ p(x | y = c) p(y = c) requires using the obtained array index to sum the values of all the columns in that row (O(C)), giving us O(D + C).

2 Sequence Prediction

Problem A

```
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

Forward algorithm	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	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

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

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

Problem E

The transition and emission matrices from 2D have a greater range of numbers along each row than those from 2C. For example, we observe a value 3.907e-09 in the transition matrix and 6.108e-15 in observation matrix from 2D, which are values very close to 0, whereas the smallest order of magnitude from matrices from 2C is -3, meaning matrices in 2D are more sparse (approximating those really small values as 0). 2C always produces the same A and O for a given training set, since there is a closed-form solution for the global optimum, while 2D produces different A's and O's every run because the Baum-Welch algorithm can only find the local optimum that varies depending on the randomly initialized A and O. Assuming the training set is a good representation of Ron's moods and their effects on his music choices, A and O from the supervised learning (2C) provide a more accurate representation, since the states directly represent his moods and the A and O are the global optima, while those from unsupervised (2D) would have hidden states that don't necessarily represent Ron's moods and are not guaranteed to be global optima. We could potentially improve the unsupervised learning by providing a meaningful initialization (if there is a prior belief, assumption, or knowledge about A and O and what the states could represent).

Problem F

File #0:
Generated Emission
##
47430172656255526245
66527575457656427125
75424565165556671136
27752153227752122243
26074471426264533147

File #1:
Generated Emission
##
02472225325055257540
27677570265027215121
61327452055023462014
60071021537214402456
42442465047027717505

File #2:
Generated Emission
##
21072160872642760957
02557515616319715522
70093466204683522797
55177522040033793230
06667407931397853172

File #3:
Generated Emission
##
23663363402622121120
30303106261666656026
60326143063256016003
50610166411161001615
11122211066420122136

File #4:
Generated Emission
##
33060561240116603664
16243525112116216332
14041123655106266422
31452550660212442624
61604434401402654113

File #5:
Generated Emission
##
02400183646560033582
20866103186046408338
83328833401881826380
86408610364512250658
36333131664604064157

Problem G

The trained A and O matrices are both sparse, but O is more sparse than A. The sparsity along each row i of A means that given a state i, only a few states are achievable in the next state (columns of intensity 0 means 0 probability of transitioning from state i to the state represented by that column). Similarly, the sparsity along each row i of O means that given a state i, only a few observations can be generated from that state (columns of intensity 0 means 0 probability of state i generating the observation represented by that column). Since O is more sparse than A, it could be interpreted as the association between states and observations being stronger than that between different states (transitioning from one to another). Some columns of O have very low intensity across all the rows, which means that the observations corresponding to those columns have overall very low probabilities of occuring (all states are unlikely to generate those observations).

Problem H

The sample emission sentences from the HMM become less nonsensical and a closer resemblance of sentences in the Constitution as the number of hidden states is increased. When there is only one hidden state, the words are arranged in a completely random order; all the states in the state sequence have the same value, so each word (observation) has the same probability of appearing anywhere in the sentence (it does not imply that all the words have the sample probability of occuring, for this probability is determined by the frequency of the word in our training set). In general, we can increase the training likelihood by allowing more hidden states since we can make more specific and stronger (sparser A, O) associations between states and words and among states; consider the extreme case where we have as many hidden states as observations, for which we could then have one word for each state, generated with probability 1 by that state, and the state transition probabilities would also be based on the frequency of a specific transition between two words from the training set, thus precisely fitting the training data and maximizing the training likelihood.

Problem I

State 2 seems to generate a lot of determiners, including quantifiers, numbers, distributives, and difference words. This state seems more semantically meaningful than others in a sense that some of the other states seem to group words together by their frequency, some others group related words in the context of the Constitution, and some others are seemingly random, whereas a lot of emissions from state 2 serve similar grammatical purposes. For example, some keywords that stand out are numbers (cardinal and ordinal), including one, two, three, thirty, second, and fourth. There are quantifiers (i.e. every, whole), distributives (i.e. every, either), and difference words (i.e. other).