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

Problem A [5 points]: Parameters of Full Model with Factorizing

Solution A.: Since there are C classes in total for y, the total number of parameters for the full model scales with C. $p(x|y=c) = \prod_{j=1}^{D} p(x_j|x_1,\ldots,x_{j-1},y=c) = \prod_{j=1}^{D} \theta_{xjc}$. Using Bayes Rule, $\prod_{j=1}^{D} \theta_{xjc} = \prod_{j=1}^{D} \frac{p(x_1,\ldots,x_j|y=c)}{p(x_1,\ldots,x_{j-1}|y=c)}$. Since each $x_j \in \{x_1,\ldots,x_D\}$ can have value 0 or 1, for each θ_{xjc}, there need to be 2^j parameters for all x_j for each θ_{xjc}. Thus, in total, there are $\sum_{j}^{D} 2^j = 2^{D+1} - 1$ parameters that need to be stored for all possible values of all the x_j's here, which is $O(2^D)$. Therefore, the total number of parameters for the full model is $O(2^D C)$.

Problem B [5 points]: Parameters of Full Model without Factorizing

Solution B.: In order to compute $p(x|y = c)$ for arbitrary x and c, since there are C classes, there is a factor of C to include here for all possible $y = c$. Since x has length D and any $x_j \in \{0, 1\}$, then there are 2^D total possible values of x. In total, there are $O(2^D C)$ parameters needed to be estimated in order to be able to compute $p(x|y = c)$. The number of parameters here is the same as in part A.

Problem C [2 points]: Naive Bayes vs. Full Model for Small N

Solution C.: If the sample size N is very small, Naive Bayes is likely to give lower test set error. This is because the full Bayes model attempts to build a model of dependencies between all the x_j in x, making the full Bayes model give a higher test error, while Naive Bayes gives a lower test error by having a simpler model that assumes less of the small dataset than full Bayes does.

Problem D [2 points]: Naive Bayes vs. Full Model for Large N

Solution D.: If the sample size N is very large, then full Bayes is likely to give lower test set error. This is because the Naive Bayes model does not account for possible dependencies between the $x_j \in x$, so Naive Bayes likely will mistakenly assume that the $x_j \in x$ are independent, while the full Bayes model instead is able to learn a more complex model that accounts for the dependencies between the $x_j \in x$.

Problem E [11 points]: Computational Complexity of Making a Prediction Using Naive Bayes vs Full Model

Solution E.: Naive Bayes: For Naive Bayes, using Bayes Rule, $p(y|x) = \frac{p(y,x)}{p(x)} = \frac{p(y)}{p(x)}p(x|y) \propto p(y)\prod_{j=1}^{D} p(x_j|y)$. Assuming a uniform class prior $p(y) = \frac{1}{C}$, $p(y)\prod_{j=1}^{D} p(x_j|y) = \frac{1}{C}\prod_{j=1}^{D} p(x_j|y)$. Thus, in order to calculate $p(y|x)$, D multiplications must occur, so the computational complexity of making a prediction using Naive Bayes for a single test case is $O(D)$.

Full Model: For the Full Model, using Bayes Rule, $p(y|x) = \frac{p(y,x)}{p(x)} = \frac{p(y)}{p(x)}p(x|y) \propto p(y)p(x|y=c)$. Assuming a uniform class prior $p(y) = \frac{1}{C}$, $p(y)p(x|y=c) = \frac{1}{C}p(x|y=c)$. Thus, in order to calculate $p(y|x)$, the value of $p(x|y=c)$ must be looked up in an array indexed here at x, which is a D-bit vector. Therefore, since converting a D-bit vector to an array index is a $O(D)$ operation, the computational complexity of making a prediction using the Full Model for a single test case is $O(D)$.

2 Sequence Prediction [75 Points]

Problem A [10 points]: Max-Probability State Sequences for 6 Trained HMMs

Solution A.: Code output:

Running Code For Question 2A

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]: Probability of Emission Sequence for 6 Trained HMMs

Solution B.: Code output for 2Bi.py

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
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

Code output for 2Bii.py

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
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

Philip Carr
February 2019

Problem C [10 points]: Learned State Transition and Output Emission Matrices of Supervised Hidden Markov Model

Solution C.: Code output:

Running Code For Question 2C

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]: Learned State Transition and Output Emission Matrices of Unsupervised Hidden Markov Model

Solution D.: Code output (excluding iteration count updates):

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

Problem E [5 points]: Compare 2C and 2D

Solution E.: The transition matrix and observation matrix in part D have more values closer to 0 than the transition matrix and observation matrix in part C, respectively, meaning that the matrices of part D are more sparse than the respective matrices in part C. The supervised model provides a more accurate representation of Ron's moods and how they affect his music choices, since the supervised model learns from training classes (y values), while the unsupervised model. It is also more realistic that the respective matrices in part C better fit Ron's behavior since there is likely a non-zero chance that Ron could switch between any two moods from one day to the next. The unsupervised model could be improved by cross-validating the current results of unsupervised training with test data to make sure that the unsupervised model stays within a realistic scope of the actual test data.

Problem F [5 points]: Generating Emission Sequences

Solution F.: Code output:

Running Code For Question 2F

File #0:
Generated Emission
77077717626564520720
05523174522726215322
35536520375156204274
76767040443441502260
03125515404175003176

File #1:
Generated Emission
16571157252356051424
65567225725704157535
25025055354244270565
67513264727453425664
57016664571425372154

File #2:
Generated Emission
10870126792525954059
29706407713432011166
73586763718157710626
92164229021361326052
66798737313753566594

File #3:
Generated Emission
21015426166053234352
66116005513036323226
11122146331431506262
26664252023123203632
52523526112633141432

File #4:
Generated Emission

36121136011032132514
43366243156666030431
15330662224204203524
14305002504561265230
04536066350254206322

File #5:
Generated Emission
26256416531228656616
88866441546464383068
07265622343323534303
01166316834166841666
03588308524830408823

Problem G [3 points]: Sparsity of Trained A and O Matrices

Solution G.:

Problem H [5 points]: Hidden States vs. Sample Emission Sentences from HMM

Solution H.:

Philip Carr

Problem I [5 points]: Analyzing Visualization of State

Solution I.: