Note: All code is in the week2 folder

Problem 1

A

The time complexity of the naive HMM algorithm is

$$O(M \cdot L^M)$$

where L is the number of states and M is the number of observations. This is because there are L^M possibilities/classes, and we just want to enumerate them (L^M) and choose the best one (M time to score each one by multiplying emission and transition probabilities over the length of the sequence).

B

The time complexity of Viterbi is

$$O(M \cdot L^2)$$

where L is the number of states and M is the number of observations. This is because we consider the best k length prefix ending in each observation (so at each length, we need to consider ending in any state, and when we consider ending in a state, we need to consider coming previously from any state).

C (code in seq_pred.py)

See code file.

D

```
FILE = sequenceprediction1.txt
MAXSTATESEQ(77550) = 22222
MAXSTATESEQ(7224523677) = 2222221000
MAXSTATESEQ(505767442426747) = 222100003310031
MAXSTATESEQ(72134131645536112267) = 10310310000310333100
MAXSTATESEQ(4733667771450051060253041) = 2221000003222223103222223
P(77550) = 0.00011810020361
P(7224523677) = 2.03268952868e-09
P(505767442426747) = 2.47736684344e-13
P(72134131645536112267) = 8.8711581199e-20
P(4733667771450051060253041) = 3.739956487e-24

FILE = sequenceprediction2.txt
MAXSTATESEQ(60622) = 11111
MAXSTATESEQ(4687981156) = 2100202111
MAXSTATESEQ(815833657775062) = 021011111111111
MAXSTATESEQ(21310222515963505015) = 02020111111111111021
MAXSTATESEQ(6503199452571274006320025) = 1110202111111102021110211
P(60622) = 2.0883778502e-05
P(4687981156) = 5.18099069122e-11
P(815833657775062) = 3.31515789374e-15
P(21310222515963505015) = 5.12595690734e-20
P(6503199452571274006320025) = 1.29696134676e-25

FILE = sequenceprediction3.txt
MAXSTATESEQ(13661) = 00021
MAXSTATESEQ(2102213421) = 3131310213
MAXSTATESEQ(166066262165133) = 133333133133100
MAXSTATESEQ(53164662112162634156) = 20000021313131002133
MAXSTATESEQ(1523541005123230226306256) = 1310021333133133313133133
P(13661) = 0.000173176766091
P(2102213421) = 8.28507920611e-09
```

```
P(166066262165133) = 1.64163879355e-12
P(53164662112162634156) = 1.06347342033e-16
P(1523541005123230226306256) = 4.53467046372e-22

FILE = sequenceprediction4.txt
MAXSTATESEQ(23664) = 01124
MAXSTATESEQ(3630535602) = 0111201112
MAXSTATESEQ(350201162150142) = 011244012441112
MAXSTATESEQ(00214005402015146362) = 11201112412444011112
MAXSTATESEQ(2111266524665143562534450) = 2012012424124011112411124
P(23664) = 0.000114136332244
P(3630535602) = 4.32621950202e-09
P(350201162150142) = 9.79337531929e-14
P(00214005402015146362) = 4.73993465995e-18
P(2111266524665143562534450) = 5.61795840291e-22

FILE = sequenceprediction5.txt
MAXSTATESEQ(68535) = 10111
MAXSTATESEQ(4546566636) = 1111111111
MAXSTATESEQ(638436858181213) = 110111010000011
MAXSTATESEQ(13240338308444514688) = 00010000000111111100
MAXSTATESEQ(0111664434441382533632626) = 2111111111111100111110101
P(68535) = 1.32174527463e-05
P(4546566636) = 2.86687793347e-09
P(638436858181213) = 4.32322750273e-14
P(13240338308444514688) = 4.62890402955e-18
P(0111664434441382533632626) = 1.43951249217e-22
```

E (code in m_training.py)

Note that for my states I used 0 = "happy", 1 = "mellow", 2 = "sad", and 3 = "angry".

Also note that for my observations I used 0 = "rock", 1 = "pop", 2 = "house", 3 = "metal", 4 = "folk", 5 = "blues", 6 = "dubstep", 7 = "jazz", 8 = "rap", and 9 = "classical".

The learned state transition matrix is as follows:

	State 0	State 1	State 2	State 3
State 0	0.2830188679245283	0.4716981132075472	0.12971698113207547	0.11556603773584906
State 1	0.2335907335907336	0.3803088803088803	0.29343629343629346	0.09266409266409266
State 2	0.1035031847133758	0.09713375796178345	0.3678343949044586	0.4315286624203822
State 3	0.18870967741935485	0.09838709677419355	0.3064516129032258	0.4064516129032258

where the rows represent initial hidden states (moods) and the columns represent subsequent hidden states (moods). So as an example, trans_mat[i][j] is the probability that Ron transitions from mood i to mood j.

The learned output emission matrix is as follows:

	Observation 0	Observation 1	Observation 2	Observation 3	Observation 4	Observation 5	Observation 6	Observation 7	Observation 8	Observation 9
State 0	0.14858490566037735	0.22877358490566038	0.15330188679245282	0.1179245283018868	0.04716981132075472	0.05188679245283019	0.02830188679245283	0.12971698113207547	0.09198113207547169	0.0023584905660377358
State 1	0.10597302504816955	0.009633911368015413	0.019267822736030827	0.030828516377649325	0.1695568400770712 8	0.046242774566473986	0.14065510597302505	0.23892100192678228	0.13872832369942195	0.1001926782273603
State 2	0.11942675159235669	0.042993630573248405	0.06528662420382166	0.09076433121019108	0.1767515923566879	0.20222929936305734	0.04617834394904458	0.050955414012738856	0.07802547770700637	0.12738853503184713
State 3	0.1693548387096774	0.03870967741935484	0.14677419354838708	0.18225806451612903	0.04838709677419355	0.06290322580645161	0.09032258064516129	0.025806451612903226	0.2161290322580645	0.01935483870967742

where the rows represent the hiddne states (moods) and the columns represent observations (music genres). So as an example, emiss_mat[i][j] is the probability that Ron is in mood i given that he is listening to genre j.

Since the above table is so small and can only be viewed if you zoom in on the PDF, I have included the raw output for your viewing pleasure.

[0.14858490566037735, 0.22877358490566038, 0.15330188679245282, 0.1179245283018868, 0.04716981132075472, 0.05188679245283019, 0.02830188679245283, 0.12971698113207547, 0.09198113207547169, 0.0023584905660377358]

[0.10597302504816955, 0.009633911368015413, 0.019267822736030827, 0.030828516377649325, 0.1695568400770712 8, 0.046242774566473986, 0.14065510597302505, 0.23892100192678228, 0.13872832369942195, 0.1001926782273603]

[0.11942675159235669, 0.042993630573248405, 0.06528662420382166, 0.09076433121019108, 0.1767515923566879, 0.20222929936305734, 0.04617834394904458, 0.050955414012738856, 0.07802547770700637, 0.12738853503184713]

[0.1693548387096774, 0.03870967741935484, 0.14677419354838708, 0.18225806451612903, 0.04838709677419355, 0.06290322580645161, 0.09032258064516129, 0.025806451612903226, 0.2161290322580645, 0.01935483870967742]

As per the clarification email, I performed 5 fold cross validation for the HMM. I got the following results.

Fold 0

Transition Matrix

[0.27976190476190477, 0.48214285714285715, 0.125, 0.1130952380952381]

[0.2289156626506024, 0.3783132530120482, 0.29156626506024097, 0.10120481927710843]

[0.108, 0.094, 0.368, 0.43]

[0.18562874251497005, 0.0998003992015968, 0.3033932135728543, 0.4111776447105788]

Emission Matrix

[0.13392857142857142, 0.2261904761904762, 0.15476190476190477, 0.12797619047619047, 0.041666666666666664, 0.0625, 0.023809523809523808, 0.12202380952380952, 0.10416666666666667, 0.002976190476190476]

[0.10096153846153846, 0.01201923076923077, 0.014423076923076924, 0.036057692307692304, 0.17067307692307693, 0.04807692307692308, 0.12980769230769232, 0.25240384615384615, 0.14182692307692307, 0.09375]

[0.124, 0.042, 0.066, 0.088, 0.17, 0.204, 0.044, 0.05, 0.08, 0.132]

[0.1656686626746507, 0.041916167664670656, 0.15169660678642716, 0.16766467065868262, 0.04790419161676647, 0.0658682634730539, 0.09181636726546906, 0.029940119760479042, 0.21956087824351297, 0.017964071856287425]

Fold error = 0.509132420091

Fold 1

Transition Matrix

[0.28059701492537314, 0.46567164179104475, 0.12835820895522387, 0.1253731343283582]

[0.2181372549019608, 0.38480392156862747, 0.30637254901960786, 0.09068627450980392]

[0.1065891472868217, 0.08914728682170543, 0.375968992248062, 0.42829457364341084]

[0.1947261663286004, 0.10141987829614604, 0.31237322515212984, 0.39148073022312374]

Emission Matrix

[0.14328358208955225, 0.2298507462686567, 0.14925373134328357, 0.11343283582089553, 0.050746268656716415, 0.050746268656716415, 0.029850746268656716, 0.1373134328358209, 0.0955223880597015, 0.0]

[0.1100244498777506, 0.007334963325183374, 0.019559902200488997, 0.029339853300733496, 0.17359413202933985, 0.05134474327628362, 0.14425427872860636, 0.22493887530562348, 0.1393643031784841, 0.10024449877750612]

[0.1182170542635659, 0.040697674418604654, 0.06976744186046512, 0.09108527131782945, 0.1744186046511628, 0.20736434108527133, 0.046511627906976744, 0.05232558139534884, 0.07945736434108527, 0.12015503875968993]

[0.1643002028397566, 0.036511156186612576, 0.15212981744421908, 0.19066937119675456, 0.0486815415821501, 0.06288032454361055, 0.09127789046653144, 0.02231237322515213, 0.21095334685598377, 0.02028397565922921]

Fold error = 0.527397260274

Fold 2

Transition Matrix

[0.2782608695652174, 0.48405797101449277, 0.1246376811594203, 0.11304347826086956]

[0.2429245283018868, 0.38207547169811323, 0.2783018867924528, 0.09669811320754718]

[0.09690721649484536, 0.10103092783505155, 0.3628865979381443, 0.43917525773195876]

[0.19678714859437751, 0.09437751004016064, 0.2971887550200803, 0.41164658634538154]

Emission Matrix

[0.15072463768115943, 0.22318840579710145, 0.15942028985507245, 0.11014492753623188, 0.057971014492753624, 0.04927536231884058, 0.02318840579710145, 0.14202898550724638, 0.08115942028985507, 0.002898550724637681]

[0.11058823529411765, 0.011764705882352941, 0.021176470588235293, 0.03058823529411765, 0.17411764705882352, 0.047058823529411764, 0.1388235294117647, 0.2376470588235294, 0.12941176470588237, 0.0988235294117647]

[0.1134020618556701, 0.04742268041237113, 0.06597938144329897, 0.09484536082474226, 0.17525773195876287, 0.18969072164948453, 0.04329896907216495, 0.05773195876288866, 0.08041237113402062, 0.13195876288659794]

[0.1606425702811245, 0.04216867469879518, 0.13855421686746988, 0.178714859437751, 0.050200803212851405, 0.0642570281124498, 0.09236947791164658, 0.02208835341365462, 0.22690763052208834, 0.024096385542168676]

Fold error = 0.522831050228

Fold 3

Transition Matrix

[0.28869047619047616, 0.44642857142857145, 0.1488095238095238, 0.11607142857142858]

[0.23192019950124687, 0.3915211970074813, 0.2892768079800499, 0.08728179551122195]

[0.10609037328094302, 0.0962671905697446, 0.3654223968565815, 0.43222003929273084]

[0.17984189723320157, 0.09090909090909091, 0.3102766798418972, 0.4189723320158103]

Emission Matrix

[0.1636904761904762, 0.23809523809523808, 0.14583333333333334, 0.125, 0.03571428571428571, 0.050595238095238096, 0.02976190476190476, 0.11904761904761904, 0.08928571428571429, 0.002976190476190476]

[0.1044776119402985, 0.007462686567164179, 0.01990049751243781, 0.029850746268656716, 0.15422885572139303, 0.03980099502487562, 0.15920398009950248, 0.236318407960199, 0.14427860696517414, 0.1044776119402985]

[0.11984282907662082, 0.03929273084479371, 0.05697445972495088, 0.09430255402750491, 0.18664047151277013, 0.20235756385068762, 0.043222003929273084, 0.05108055009823183, 0.08055009823182711, 0.12573673870333987]

[0.17588932806324112, 0.041501976284584984, 0.1442687747035573, 0.18379446640316205, 0.05138339920948617, 0.05928853754940711, 0.09090909090909091, 0.02766798418972332, 0.20948616600790515, 0.01581027667984189б]

Fold error = 0.529680365297

Fold 4

Transition Matrix

[0.2877906976744186, 0.4796511627906977, 0.11918604651162791, 0.11337209302325581]

[0.24528301886792453, 0.36556603773584906, 0.3018867924528302, 0.08726415094339622]

[0.0998003992015968, 0.10578842315369262, 0.3652694610778443, 0.4291417165668663]

[0.18672199170124482, 0.10580912863070539, 0.3112033195020747, 0.3962655601659751]

Emission Matrix

[0.1511627906976744, 0.22674418604651161, 0.1569767441860465, 0.11337209302325581, 0.04941860465116279, 0.046511627906976744, 0.03488372093023256, 0.12790697674418605, 0.09011627906976744, 0.0029069767441860465]

[0.10377358490566038, 0.009433962264150943, 0.02122641509433962, 0.02830188679245283, 0.17452830188679244, 0.04481132075471698, 0.1320754716981132, 0.2429245283018868, 0.1391509433962264, 0.10377358490566038]

[0.12151394422310757, 0.045816733067729085, 0.06772908366533864, 0.08565737051792828, 0.17729083665338646, 0.20717131474103587, 0.053784860557768925, 0.043824701195219126, 0.0697211155378486, 0.12749003984063745]

[0.18049792531120332, 0.03112033195020747, 0.14730290456431536, 0.1908713692946058, 0.043568464730290454, 0.06224066390041494, 0.08506224066390042, 0.026970954356846474, 0.21369294605809128, 0.01867219917012448]

Fold error = 0.546697038724

OVERALL CROSS VALIDATION ERROR = 0.527147626923

2

F (code in crf.py, which uses seq_pred_mod.py)

Note that for my states I used 0 = “happy”, 1 = “mellow”, 2 = “sad”, and 3 = “angry”.

Also note that for my observations I used 0 = “rock”, 1 = “pop”, 2 = “house”, 3 = “metal”, 4 = “folk”, 5 = “blues”, 6 = “dubstep”, 7 = “jazz”, 8 = “rap”, and 9 = “classical”.

Using a learning rate of .0001 and using a stopping point of .001 (meaning I stop when no value in my transition or emission matrices changes by more than .001) I got the following transition and emission scoring matrices when training on all the data. I got these model parameters after 338 epochs, meaning both matrices had stabilized by the 339th epoch. Note that I initialized all the values using a uniform random distribution between 0 and 1.

Transition Scoring Matrix

	State 0	State 1	State 2	State 3
State 0	0.8487675078175694	0.7091787678678413	-0.04004272763004517	0.20575367041702378
State 1	1.2783453054864864	1.0766920795611732	-0.23331340350355875	-0.5005326465024897
State 2	0.15342011907989433	0.9078833683214806	1.1514249058648907	0.6893681913901086
State 3	0.05403674366979719	-0.09447173277421665	1.3501963497108085	1.0080509476912538

where the columns represent initial hidden states (moods) and the rows represent subsequent hidden states (moods).

Emission Scoring Matrix

	Observation 0	Observation 1	Observation 2	Observation 3	Observation 4	Observation 5	Observation 6	Observation 7	Observation 8	Observation 9
State 0	0.678576129709364	1.5660195491974152	1.0866066699254782	0.4891073909266601	-0.026627156928609834	0.22001582041876286	-0.31228565923031115	0.7207442556082864	0.22039626545581772	0.09824653352323524
State 1	0.47157349092503487	-0.32043398464972866	-0.10183429520303876	-0.12782008160023584	1.0987071547053429	0.2893557381096954	0.7303341053206283	1.433684170629346	0.6243096725263724	1.1985808872324093
State 2	0.4300309427002712	0.15451513204335834	0.15109866059897975	0.210943678471252	1.170838309186604	1.324582866464665	-0.14365718030619032	0.29413661835517	-0.07361586021268972	1.480897128796533
State 3	0.9813684811872891	0.3243694857402104	1.1555949685549278	1.0550964808181433	0.36214883345390125	0.48985328428945285	0.49748203454144985	0.1448778347519093	1.1300812245578384	0.3211769587341272

where the rows represent the hidden states (moods) and the columns represent observations (music genres).

Since the above table is so small and can only be viewed if you zoom in on the PDF, I have included the raw output for your viewing pleasure.

[0.678576129709364, 1.5660195491974152, 1.0866066699254782, 0.4891073909266601, -0.026627156928609834, 0.22001582041876286, -0.31228565923031115, 0.7207442556082864, 0.22039626545581772, 0.09824653352323524]

[0.47157349092503487, -0.32043398464972866, -0.10183429520303876, -0.12782008160023584, 1.0987071547053429, 0.2893557381096954, 0.7303341053206283, 1.433684170629346, 0.6243096725263724, 1.1985808872324093]

[0.4300309427002712, 0.15451513204335834, 0.15109866059897975, 0.210943678471252, 1.170838309186604, 1.324582866464665, -0.14365718030619032, 0.29413661835517, -0.07361586021268972, 1.480897128796533]

[0.9813684811872891, 0.3243694857402104, 1.1555949685549278, 1.0550964808181433, 0.36214883345390125, 0.48985328428945285, 0.49748203454144985, 0.1448778347519093, 1.1300812245578384, 0.3211769587341272]

Per the update email, I performed 5-fold cross validation on my CRF model. I got the following results: **Cross validation error = 0.555903308682** The error for each slice was 0.54337899543379, 0.5616438356164384, 0.5114155251141552, 0.591324200913242, and 0.571753986332574, and each slice ran for around 400 epochs (very rough).

We can see that our cross validation error for CRFs is higher than our cross validation error for HMMs. This means that the CRF does not outperform the HMM. However, under certain conditions, a CRF is better applicable than an HMM. With CRFs, it is easier to model more complex, non-linear models, because the feature functions of CRFs can incorporate arbitrary features. For example, CRFs can take word context into account. Also CRFs don't rely as much on independence assumptions, unlike HMMs. Thus, when trying to take into account more complex features, or trying to come up with a more complex model, CRFs are more applicable than HMMs.

G

True. This is true because if we increase the number of hidden states, we increase the number of possible state sequences that are available to choose from. And if we have more possible state sequences to choose from, it becomes more likely that we can exactly match the training data. More precisely, it is more likely that given more states, we can associate each point in the observation sequence with a correct hidden state. It's basically analagous to increasing model complexity in order to increase the chances of exactly fitting the training data.

H

In the EM algorithm, we update our transition matrix during the maximization step. We update the values for each cell by summing over marginals. The marginals are computed by multiplying alpha values (computed via the forward algorithm) and beta values (computed via the backward algorithm).

So basically, we just need to show the marginals are zero in the specified cases.

If a coefficient of the initial state probability matrix is initially 0, then that coefficient will remain 0 until the end of the EM algorithm. This is because in the forward algorithm, the value for that row in the forward algorithm will always remain 0. This is because the initial probability of having that state for any observation is 0 (and these probabilities don't get updated, as we prove), so when you do the forward algorithm and you sum the values of the previous columns, the emission probabilities will always be 0. Thus, when you update the matrix during the maximization step by summing the marginals, nothing happens to the coefficents that were initially zero, because the alpha values (computed using the forward algorithm) and thus the marginals for those coefficients is 0.

Notice that if a coefficent for a given state of the state transition probability matrix is 0, then that coefficient will remain 0 until the end of the EM algorithm. This is because in the forward algorithm, the value for that row in the forward matrix will always remain 0. This is because the initial probability of transitioning to that state is 0 (and this probability doesn't get updated, as we prove). So when you sum the values of transitioning from that state from all the states in the previous column, you just get 0. Thus, when you update the transition matrix during the maximization step by summing the marginals, nothing happens to the coefficients that were initially zero, because the alpha values (computed using the forward algorithm) and thus the marginals for those coefficients is 0.