Question A&B:
(Implemented in CRF.py and pre_mod.py)

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: (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).

Question C:

Question D: The Optimality Condition requires that the gradient be 0 for the optimal w, which implies (assuming we're not regularizing w):

$$\sum_{j=1}^{M}\sum_{i=1}^{N}\phi^j(y_i^j, y_i^{j-1}|\mathbf{x}_i) = \sum_{j=1}^{M}\sum_{i=1}^{N}E_{\mathbf{y}'\sim P_w(\mathbf{Y}|\mathbf{x}_i)}[\phi^j(y'^j, y'^{j-1}|\mathbf{x}_i)].$$

In other words, for the optimal w, the expected value of each feature $\phi^j(y'^j, y'j-1|x)$ over the conditional probability $P_w(Y = y'|x)$ should equal the frequency counts of that feature over the training set. In the case where each $\phi^j(y'^j, y'j-1|x)$ occupies a disjoint region of the total feature vector:

$$\forall j \in \{1,\ldots,M\}: \sum_{i=1}^{N}\phi^j(y_i^j, y_i^{j-1}|\mathbf{x}_i) = \sum_{i=1}^{N}E_{\mathbf{y}'\sim P_w(\mathbf{Y}|\mathbf{x}_i)}[\phi^j(y'^j, y'^{j-1}|\mathbf{x}_i)].$$

Question D: 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.