Homework 5
HMM, Model Selection and Regularization

CMU 10-601: Machine Learning (Fall 2015)
http://www.cs.cmu.edu/~10601b/
OUT: Oct. 27, 2015
DUE: Nov 10, 2015, 10:30 AM

START HERE: Instructions

- The homework is due at 10:30 am on Tuesday November 10, 2015. Each student will given three late days that can be spent on any homeworks but not on projects. Once you have used up your late days for the term, late homework submissions will receive 50% of the grade if they are one day late, and 0% if they are late by more than one day.
- ALL answers will be submitted electronically through the submission website: https://autolab.cs.cmu.edu/courses/10601b-f15/assessments. You can sign in using your Andrew credentials. You should make sure to edit your account information and choose a nickname/handle. This handle will be used to display your results for any competition questions (such as the class project) on the class leaderboard.
- All questions will be *autograded*. Please make sure to carefully follow the submission instructions for these questions. A template submission can be downloaded at http://www.cs.cmu.edu/~10601b/assignments/hw5template.tar
- Collaboration on solving the homework is allowed (after you have thought about the problems on your own). When you do collaborate, you should list your collaborators! You might also have gotten some inspiration from resources (books or online etc...). This might be OK only after you have tried to solve the problem, and couldn't. In such a case, you should cite your resources.
- If you do collaborate with someone or use a book or website, you are expected to write up your solution independently. That is, close the book and all of your notes before starting to write up your solution. You should also state your collaborations in your short-answer writeup. Specifically, please write down the following:

  1. Did you receive any help whatsoever from anyone in solving this assignment? Yes / No. If you answered yes, give full details: (e.g., "Jane explained to me what is asked in Question 3.4").
  2. Did you give any help whatsoever to anyone in solving this assignment? Yes / No. If you answered yes, give full details: (e.g., "I pointed Joe to section 2.3 to help him with Question 2").

  Collaboration without full disclosure will be handled severely, in compliance with CMU's Policy on Cheating and Plagiarism.

1 Short Answer Question [Pengcheng Zhou; 30 points]

- **Submission Instructions:** For each question you will be required to write your answers in a single text file. This file needs to have the same name as the question, and may have the extension .txt. For example, "1.1" or "1.1.txt". On each line, write your answer **and only your answer, no computations**, as a single floating point number, to at least 3 decimal places. You may number the lines of the file with the **question number and a period** if you like, for example "1. <answer>", but be consistent.

1.1 Hidden Markov Model (HMM) [21pts]

1. Assume that we have a HMM with 10 different states and a total of 100 possible outputs across all states, how many parameters are required to fully define this HMM? **[3pts]**

2. For a HMM with 3 states A, B, C, its transition probabilities are shown in Table (1). Find the initial probabilities of three states $[P(q_1 = A), P(q_1 = B), P(q_1 = C)]$ such that $P(q_2 = S_i) = P(q_1 = S_i)$, i.e., the distribution of three states is stable. (write your solution in the fomat of **a, b, c**). **[3pts]**

3. Suppose that we have binary states (labeled A and B) and binary observations (0 and 1). The initial, transition and emission probabilities are shown in Table (2)(3)(4) respectively. **[15pts]**

a. As shown in Table (2), the probability of starting from state A is $P(q_1 = A) = 0.6$. Compute $P(q_3 = A)$ (**round to 4 decimal places, same for all the following questions**). **[2pts]**

b. What is the probability of the state sequence ABA? **[2pts]**

c. Suppose that we observe the sequence $o_1 = 1, o_2 = 0$ and $o_3 = 1$, compute $\alpha_2(A)$ and $\alpha_2(B)$ which is defined as $\alpha_t(i) = P(o_1, o_2, \cdots o_t \hat{} \ q_t = S_i)$. **[2pts]**

d. What is the probability of observing sequence $o_1 = 1, o_2 = 0$ and $o_3 = 1$, $P(o_1 = 1, o_2 = 0, o_3 = 1)$? **[3pts]**

e. Given the sequence $o_1 = 1, o_2 = 0$ and $o_3 = 1$, what is the probability of $q_3 = A$? In other words, compute $P(q_3 = A|o_1 = 1, o_2 = 0, o_3 = 1)$. **[2pt]**

f. Use the Viterbi algorithm to compute the most likely sequence of states given the observations $o_1 = 1, o_2 = 0$. (show your results in the form of **XX**). **[4pt]**

1.2 Model Selection, Regularization [9pts]

1. We can do model selection by maximizing either AIC or BIC, where $AIC = \ln L - k, BIC = \ln L - \frac{k}{2} \ln N$, L is the maximum likelihood of the data using the specific model, k is the number of parameters and N is the number of data points. Which criterion achieves simpler model (smaller k) when $N > 10$? **[3pts]**

	$S_1 = A$	$S_1 = B$	$S_1 = C$
$S_2 = A$	0.7	0.2	0.1
$S_2 = B$	0.1	0.6	0.1
$S_2 = C$	0.2	0.2	0.8

Table 1: Transition Probabilities

State	$P(S_1)$
A	0.6
B	0.4

Table 2: Initial Prob.

S_2	S_1	$P(S_2\|S_1)$
A	A	0.7
A	B	0.3
B	A	0.3
B	B	0.7

Table 3: Transition Prob.

S	O	$P(O\|S)$
A	0	0.8
A	1	0.2
B	0	0.1
B	1	0.9

Table 4: Emission Prob.

2. In the lecture notes, we have shown that the prior in maximum a posterior probability (MAP) estimation is equivalent to the regularization term in penalized likelihood $\lambda\|\theta\|$. Suppose the prior of β is $\beta \sim N(0, \sigma^2 I)$, where I is the identity matrix and $\sigma = 5$, what is the corresponding λ in l_2 penalty term $\lambda\|\beta\|_2^2$? **[3pts]**

3. If β has a Laplace prior, i.e., $P(\beta_i) \propto e^{-|\beta_i|/t}$, what is the value of λ in the l_1 penalty term $\lambda\|\beta\|_1$ when $t = 2$? **[3pts]**

2 Programming [Joe Runde; 70 points]

- **Octave:** You must write your code in Octave. Octave is a free scientific programming language, with syntax identical to that of MATLAB. Installation instructions can be found on the Octave website. (You can develop your code in MATLAB if you prefer, but **you must test it in Octave** before submitting, or it may fail in the autograder.)
- **Autograding:** This problem is autograded using the CMU Autolab system. The code which you write will be executed remotely against a suite of tests, and the results used to automatically assign you a grade. To make sure your code executes correctly on our servers, you should avoid using libraries which are not present in the basic Octave install.
- **Submission Instructions:** For each sub-question you will be given a single function signature. You will be asked to write a single Octave function which satisfies the signature. In the code handout linked at the top of the assignment, we have provided you with a single folder containing stubs for each of the functions you need to complete. Do not rename these files. Complete each of these functions, then compress the files as a tar archive and submit to Autolab online. You may submit code as many times as you like. When you download the files, you should confirm that the autograder is functioning correctly by compressing and submitting the directory of stubs provided. This should result in a grade of zero for all questions.
- **READ THIS:**
  When writing your program, ensure that it both accepts input matrices and outputs matrices with the dimensions given in the problem. If it doesn't, **you will not receive any points.**

2.1 Hidden Markov Models (HMM) [70 pts]

In this question you will implement the Baum-Welch algorithm for HMM. The dataset for testing these functions can be downloaded at http://www.cs.cmu.edu/~10601b/assignments/hw5data.mat. You should be able to load it this time with:

```
load hw5data.mat
```

This file contains a, b, p, X, and $Xtest$. These parameters happen to describe a Bayesian Knowledge Tracing model, a setting of HMM that describes student learning of a topic. In this setting, there are two states, as the student has either mastered, or not mastered, the topic. a describes the student's transitions between these states. From each of these states, the student has a different probability of answering a question correctly. b describes these emission probabilities, and X is a list of trajectories per student, each showing the student's correct or incorrect answer on a series of questions.

To submit, tar the .m files (along with your written files). Do not include any .mat files, and do not tar the directory itself.

The input data for this problem is:

- X is an $N \times T$ training data matrix. Each row is an observed sequence from the HMM, with length T.
- $Xtest$ is an $N' \times T$ testing data matrix.
- (Whenever mentioned, y is the hidden state.)
- K is the number of hidden states.
- M is the number of possible observations
- a is the $K \times K$ transition matrix, where a_{ij} is the transition probability from state i to state j.
- b is the $K \times M$ emission matrix. b_{ij} is the probability to emit observation j from state i.
- p is the $K \times 1$ initial probability for starting states. $p_k = P(y_1 = k)$.

2.1.1 Inference [24 pts]

The outputs of the inference step are:

- α is a $T \times K$ matrix, where $\alpha_{t,i} = P(x1, ..., x_t, y_t = i|a, b, p)$
- A is an $N \times 1$ cell array of αs computed for each training sequence.
- β is a $T \times K$ matrix, where $\beta_{t,i} = P(x_{t+1}, ..., x_T|y_t = i, a, b, p)$
- B is an $N \times 1$ cell array of βs computed for each training sequence.

1. **Forward Computation [12 pts]**
   Complete the function [A] = Forward(a, b, p, X). This function implements the forward algorithm for each training example, placing the α matrix learned for each sequence into a cell array.

2. **Backward Computation [12 pts]**
   Complete the function [B] = Backward(a, b, p, X). This function implements the backward algorithm for each training example, placing each β matrix learned for each sequence into a cell array.

2.1.2 Learning [36 pts]

The intermediate results of the learning step are:

- γ is a $T \times K$ matrix of expected state probabilities at each time step. γ_{ti} is the probability of being in state i at time t
- Γ is an $N \times 1$ cell array, of γs computed for each training sequence.

- ξ is a $T \times K \times K$ tensor of expected transition probabilities at each time step. $\xi_{t,i,j}$ is the probability of transitioning from state i at $t-1$ to state j at time t.
- Ξ is an $N \times 1$ cell array, of ξs computed for each training sequence.

1. **E step[14 pts]**
   Complete the function [Γ, Ξ] = E_step(a, b, p, X). This function estimates the state and transition probabilities at each time step: γ and ξ, respectively. $\gamma_{ti} = P(y_t = i|X, a, b, p)$, $\xi_{t,i,j} = P(y_{t-1} = i, y_t = j|X, a, b, p)$. These are calculated for each training example, and placed into a cell array.

2. **M step[14 pts]**
   Complete the function [a, b, p] = M_step(Γ, Ξ, X, M, K). This function estimates the maximum likelihood parameters for a, b, and p, given training sequences and expected state and transition probabilities under the previous parameters.

3. **EM [8 pts]**
   Complete the function [a, b, p] = EM_estimate(a, b, p, X, nIter). This function runs the EM algorithm for nIter steps, returning updated model parameters.

2.1.3 Scoring/Generating [10 pts]

1. **Scoring Sequences[5 pts]**
   Complete the function [score] = likelihood(a, b, p, Xtest). This function calculates the log likelihood of the sequences in Xtest, returning an $N' \times 1$ matrix.

2. **Generating Sequences[5 pts]**
   Complete the function [x] = generate(a, b, p, T). This function generates one $1 \times T$ sequence of observations from the model parameters.