Homework 5
Graphical models

CMU 10-701: Machine Learning (Fall 2016)
https://piazza.com/class/is95mzbrvpn63d
OUT: November 14th
DUE: November 28th, 11:59 PM

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 3.4"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.
- **Late Submission Policy:** Late submissions will not receive full credit. Half credit will be awarded to correct solutions submitted within 48 hours of the original deadline. Otherwise, no credit will be given.
- **Submitting your work:** Non-programming parts of the assignment should be submitted as PDFs using Gradescope unless explicitly stated otherwise. Each derivation/proof should be completed on a separate page. Submissions can be handwritten, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in LaTeX. Upon submission, label each question using the template provided.

Problem 1: Bayesian Networks (Devendra) [30 points]

Matt asked Petar and Hyun Ah to grade the midterm. Petar and Hyun Ah played a game of poker to decide who will grade the midterm. Let W denote the event that Petar won the game, which depends on two binary variables, S, indicating whether Petar is skilled or not, and C, indicating whether Petar cheated or not. The probability that Petar is skilled is 0.7, while Petar cheats with probability of 0.9. The probability that Petar wins, given the values of C and S, is the following:

$$P(W = 1|C = 1, S = 1) = 0.9,\ P(W = 1|C = 1, S = 0) = 0.8$$
$$P(W = 1|C = 0, S = 1) = 0.7,\ P(W = 1|C = 0, S = 0) = 0.0$$

Now, Matt asked Hemank who won the game. Let H denote the variable indicating whether Hemank reports to Matt that Petar won the game. Hemank is Petar's friend, so he might lie to Matt if Petar loses the game. Let's say, the probability $P(H = 1|W = 0) = 0.1$ and $P(H = 1|W = 1) = 1$.

Note that all variables are binary.

1. (4 pts) Draw a Bayesian network with variables W,C,S and H.
2. (2 pts) What is the probability that Hemank reports that Petar won, given that Petar won and he cheated?
3. (2 pts) What is the probability that Hemank reports that Petar won, given that Petar won and he is skilled?
4. (2 pts) Compare the values you obtained in (b) and (c). Are they equal? If no, which value is bigger? Why are they equal or why is one bigger than the other? Explain in terms of properties of Bayesian networks.
5. (8 pts) What is the probability that Petar is skilled given that Hemank reports that Petar won?
6. (8 pts) What is the probability that Petar is skilled given that Hemank reports that Petar won and Petar cheated?
7. (4 pts]) Compare the values you obtained in (d) and (e). Are they equal? If no, which value is bigger? Why are they equal or why is one bigger than the other? Explain in terms of properties of Bayesian networks.

Show all the steps in calculating the above probabilities.

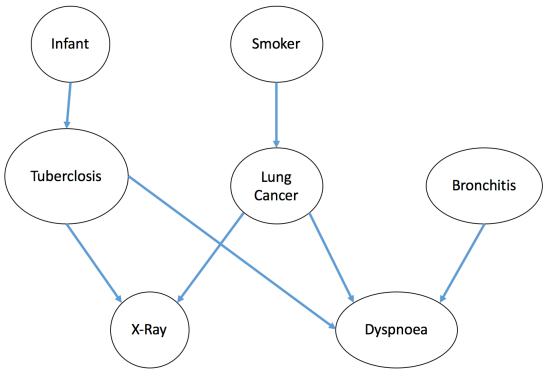

Figure 1: Graphical Model

Problem 2: Variable Elimination and Message Passing (Hemank) [10 points]

Consider the Bayesian network you generated in Problem 1. We will be trying to answer the following queries using the inference techniques like Variable Elimination. Consider the following query:

- P(Hemank Reports Petar Won | Petar is Skilled i.e. P(H=1 | S=1)

Variable Elimination

1. Draw moralized graph for the given graphical model.
2. Choose a good elimination ordering and write down the variable elimination procedure. Draw the intermediate factors produced after eliminating each variable. You need not substitute any numbers.

Another Graphical Model

Consider the graphical model shown in the Figure 1. For the given graphical model, we will use the following symbols - I (Infant), S (Smoker), T (Tuberculosis), L (Lung Cancer), B(Bronchitis), X(X-Ray), D (Dyspnoea). All of these variables are binary variables i.e. they take only 0 or 1 as values.

1. For the given graphical model, draw the moralized graph.
2. For the query P(D = 1 | S = 1), write down the variable elimination procedure. Show the steps for the variable elimination algorithm., write down the variable elimination procedure. Draw the intermediate factors for the process.
3. Comment on the time and space complexity of the variable elimination algorithm.

Problem 3: Markov Random Fields (Devendra) [25 points]

In this question we design a very simple undirected graphical model for denoising an image. For simplicity, let's assume that the image consists of only binary pixels, i.e. black or white. We have observed data $X \in [-1, 1]^{m \times n}$, where 1 denotes a black pixel while -1 denotes a white pixel. We represent the unobserved true output (i.e. the denoised image) for an $m \times n$ image as $Y \in [-1, 1]^{m \times n}$. We want to denoise the image by minimizing the difference between the observed output, as well as minimizing the difference between adjacent pixels in the predicted denoised image. We also expect some bias towards white pixels in the predicted image.

We parametrize the distribution as using the following potential functions:

$$\begin{aligned}
\psi_\alpha(Y_{ij}) &= \alpha^{\mathbb{I}\{Y_{ij}=1\}} && i \in [1, m], j \in [1, n] \\
\psi_\beta(Y_{ij}, Y_{i,j+1}) &= \beta^{\mathbb{I}\{Y_{ij}=Y_{i,j+1}\}} && i \in [1, m], j \in [1, n-1] \\
\psi_\gamma(Y_{ij}, Y_{i+1,j}) &= \gamma^{\mathbb{I}\{Y_{ij}=Y_{i+1,j}\}} && i \in [1, m-1], j \in [1, n] \\
\psi_\theta(Y_{ij}, X_{ij}) &= \theta^{\mathbb{I}\{X_{ij}=Y_{ij}\}} && i \in [1, m], j \in [1, n]
\end{aligned}$$

Here, $\mathbb{I}$ denotes the indicator function and i and j are pixel iterators.

1. (3pts) Draw the corresponding undirected graphical model for $m = 3, n = 3$.
2. (2pts) For each of the parameters $(\alpha, \beta, \gamma, \theta)$, based on your intuition state whether each is probably < 1 or > 1.
3. (5pts) Suppose we formulate this as a CRF where we condition on $\mathbf{X}$. Write out the partition function that corresponds to $P(\mathbf{Y}|\mathbf{X}, \alpha, \beta, \gamma, \theta)$.
4. (6pts) Let $E_1 = \log(p(Y_{rs} = 1, \mathbf{Y}_{-rs}|\mathbf{X}, \alpha, \beta, \gamma, \theta)$ and $E_2 = \log(p(Y_{rs} = -1, \mathbf{Y}_{-rs}|\mathbf{X}, \alpha, \beta, \gamma, \theta)$. Write the expression for $E_3 = E_1 - E_2$. Show that if $\mathbf{Y}_{-rs}, \mathbf{X}$ are equal in both E_1 and E_2, then E_3 is dependent only on pixels local to Y_{rs}. Here, $r \in [2, m-1], s \in [2, n-1]$.
5. (9pts) Suppose

$$X = \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix}, \quad \text{and } Y = \begin{bmatrix} -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix}.$$

   Write a simple program (any programming language is fine) to calculate the conditional data likelihood $(P(\mathbf{Y}|\mathbf{X}, \alpha, \beta, \gamma, \theta))$. Calculate the conditional likelihood for the following parameter settings using your program:

   (a) $\alpha = 0.9, \beta = 1.1, \gamma = 0.8, \theta = 0.9$

   (b) $\alpha = 0.8, \beta = 1.5, \gamma = 1.2, \theta = 1.0$

   State which parameter setting leads to a greater conditional data likelihood. Please include your code as an appendix in the PDF report.

Problem 4: Hidden Markov Model: Implementation (Siddharth) [30 points]

In this problem you will be implementing a Hidden Markov Model (HMM) for the task of parts-of-speech (POS) tagging. Please download the zip file containing the required files for this question from *here*.

Introduction

An HMM represents probability distributions over sequences of observations. Let Y_t denote a random variable corresponding to an observation at time t (and the observation value is y_t).

An assumption in the HMM is that the observation at time t is generated by some process whose state X_t is hidden. Moreover, the hidden process is assumed to satisfy the Markov property, which means that the current state (X_t) given previous state (X_{t-1}) is independent of all states before time $t-1$. The third assumption in the HMM is that the hidden state is discrete.

For the task of POS tagging, each Y_t would take one of the words or unigram symbols ({'the', 'homework', 'model' ... }), and X_t would be one of the POS tags ({'N' (noun), 'R' (adverb), ... }).

The resulting joint probability distribution of the hidden states and the observations can be then written as:

$$P(Y_{1:T}, X_{1:T}) = P(X_1)P(Y_1|X_1)\prod_{t=2}^{T}(P(X_t|X_{t-1})P(Y_t|X_t)) \tag{1}$$

where $Y_{1:T}$ is the sequence of T observation random variables, $Y_1, Y_2 ... Y_T$.

In order to completely specify this joint distribution, we need 3 things:

1. probability distribution over initial states $P(X_1)$ (also known as the prior probabilities)
2. state transition probability matrix $P(X_t|X_{t-1})$
3. emission probability distribution $P(Y_t|X_t)$.

Another assumption in the HMM is that the state transition probabilities and the emission probabilities are time-invariant (so there is no dependence on t).

If we consider N words (number of all possible observations = N), and M states (the total number of hidden states = M), then we need:

1. M parameters for specifying the prior distribution over states. Let's call this probability vector as π, where, π_i is the probability that the starting state is x_i. To be more specific (for this problem), each x_i corresponds to a POS tag. All unique POS tags are in the file 'POStags.txt'. (Please go to file details section to know about the exact meaning of these tags).
2. $M \times M$ parameters for the time-invariant transition probability matrix (S). Let's denote this probability matrix by S, with $S_{i,j}$ storing the transition probability from state x_i to x_j.
3. $M \times N$ parameters for the emission probabilities. Let's call this matrix as E, where $E_{i,k}$ is the emission probability of word y_k from state x_i. All unique words are in the file 'allsymbols.txt'.

These are the 3 sets of parameters in an HMM (let θ denote all the parameters, $\theta = \{\pi, S, E\}$).

The 3 tasks associated with an HMM are:

1. Learning: Obtaining the appropriate setting of parameters by maximizing the probability of generating observed sequences.
2. Evaluation: Evaluating the probability of an observed sequence of word ($y_{1:T}$), given the set of HMM parameters.

3. Decoding: Finding the most likely sequence of hidden states, associated with an observed sequence (and given the set of HMM parameters).

In this assignment, you will implement forward and backward algorithms for the task of evaluation, and Viterbi algorithm for the task of decoding.

Evaluation: Forward and Backward algorithm

For the task of evaluation, suppose we are given a sequence of observations $y_{1:T}$. We wish to compute the probability of the observed sequence given the HMM parameters.

Forward algorithm

The goal of the forward algorithm is to compute the joint probabilities $\alpha_{t,i} = P(Y_{1:t} = y_{1:t}, X_t = x_i)$. $\alpha_{t,i}$ basically stands for the probability that all words have been generated up to (and including) time t, and the current state at time t is x_i. The α values for each state and each time, can be computed by the following recursive formulation:

- Base case: $\alpha_{1,i} = \pi_i E_{i,1}$ (Note $E_{i,1}$ = emission probability of word y_1 from state x_i)
- for $t = 2, \cdots, T$: $\alpha_{t,i} = E_{i,t} \times \sum_{k=1}^{M} (\alpha_{t-1,k} \times S_{k,i})$

Note: you need to repeat this for all M states (look at the index i). So essentially, α is a $M \times T$ matrix.

The probability of the observed sequence given the parameters can now be written as:

$$P(Y_{1:T} = y_{1:T}|\theta) = \sum_{i=1}^{M} \alpha_{T,i} \tag{2}$$

Backward algorithm

The goal of the backward algorithm is to compute backward probabilities β. The backward probability intuitively is the probability of generating all words after time t, given that at time t the state is x_i. Formally, $\beta_{t,i} = P(Y_{t+1:T} = y_{t+1:T}|X_t = x_i, \theta)$. The backward probabilities can be computed in a similar recursive manner:

- Base case: $\beta_{T,i} = 1$.
- for $t = T, \cdots, 2$: $\beta_{t-1,i} = \sum_{k=1}^{M} (\beta_{t,k} \times S_{i,k} \times E_{k,t})$

Note: As we saw previously, you need to repeat this for all M states (look at the index i). So essentially, β is a $M \times T$ matrix.

The probability of the observed sequence using the β's can be then written as:

$$P(Y_{1:T} = y_{1:T}|\theta) = \sum_{i=1}^{M} (\pi_i \times E_{i,1} \times \beta_{1,i}) \tag{3}$$

File details

The dataset that we are going to use is taken from `http://www.cs.cmu.edu/~ark/TweetNLP/`. The meanings of the POS tags can be found in Table 6 of *this paper*.

The zip files contains the files for:

1. prior probabilities ('prior.txt'): Each line contains a POS tag, and the prior probability (joined by ':'). The format is:

```
<postag1>:0.01
<postag2>:0.2
.
.
.
<postagM>:0.5
```

2. emission probabilities ('emission.txt'): Each line contains a POS tag as the first word, it is followed by list of pairs of words and probabilities (which are joined together using "%"). More specifically: the format is:

```
<postag1> <w1>%0.01 <w2>%0.02  ...
<postag2> <w1>%0.002 <w2>%0.1 ...
.
.
.
<postagM> <w1>%=0.01 <w2>%0.003
```

3. transition probabilities ('transition.txt'): Each line contains a POS tag as the first word, and it is followed by a list of pairs of POS tags and probabilities (which are joined together using ":"). The format is:

```
<postag1> <postag1>:0.01 <postag2>:0.02  ...  <postagM>:0.001
<postag2> <postag1>:0.0001 <postag2>:0.2  ...  <postagM>:0.1
.
.
.
<postagM> <postag1>:0.003 <postag2>:0.03  ...  <postagM>:0.001
```

You need to implement both the forward and backward algorithms for the tweets given in file 'testsym.txt' (the file has one tweet per line, in every line, each word is separated by a space character). To avoid numerical underflow, you must do all operations in log space.

Please report the log probability of generating each of the tweet. You should obtain same log probabilities for the sequences from both the forward and the backward algorithms for all the sequences (TASK1).

Decoding: Viterbi algorithm

In order to find the most likely sequence of states given an observed sequence (and the set of parameters), we will use the Viterbi algorithm.

Let $h_{1:T} = h_1, h_2...h_T$ be the sequence of hidden states, we wish to find $\text{argmax}_h P(H_{1:T} = h_{1:T}|y_{1:T}, \theta)$ given a observed sequence $y_{1:T}$ of length T. The Viterbi algorithm is a dynamic programming algorithm, that keeps track of the most likely path ending at each of the M different states at time t.

Let $V_{t,i}$ store the most likely state sequence for the label sub-sequence $y_{1:t}$, ending at state x_i. So at the end, we will have V as a matrix of size $T \times M$, where each entry will be a list of states. Let $B_{t,i}$ be the probability of the state sequence $V_{t,i}$ and generating the sub-sequence $y_{1:t}$. The B matrix will also be a matrix of size $T \times M$, with each entry storing probability values. The Viterbi algorithm can be written as:

1. Base case:
   - $B_{1,i} = \pi_i E_{i,1}$
   - $V_{1,i} = [x_i]$ (note this is a list with the tag x_i as an element)
2. for $t = 2, \cdots, T$:

- $B_{t,i} = \max_{1 \leq k \leq M}(B_{t-1,k} \times S_{k,i} \times E_{i,t})$
- $V_{t,i} = V_{t-1,k} + [x_i]$, (note this is list concatenation), where $k = \text{argmax}_{1 \leq k \leq M} B_{t-1,k} \times S_{k,i} \times E_{i,t}$

Finally, the most likely state sequence $h^* = V_{T,k}$, where $k = \text{argmax}_{1 \leq k \leq M} B_{T,k}$.

Run your implementation to find the POS tags for the tweets given in file 'testsym.txt'. Report the 5 sequences of POS tags. (TASK2)

Problem-specific details

- Please implement all the 3 algorithms from scratch. Using any third party library or already existing implementation is not allowed. (If you are coding in Python, using Numpy is fine).
- You must do all operations in log space, which means that you need to worry about efficiently implementing operations of the form $\log(a + b)$.
- You should also attach your code with the report (in addition to submitting in Autolab).
- Just to reiterate, if you were confused about what has to be done, the 2 tasks for this problem can be searched using TASK keyword.