CS1675: Homework 9

Due: 12/9/2018, 11:59pm

This assignment is worth 40 points.

Part I: Bayes net written exercises (20 points)

Enter your responses in a file `report.pdf/docx` .

1. [10 pts] Bishop Exercise 8.11
2. [10 pts] In this exercise, we'll do some cross-domain recommendation, where we assume that there is a correlation between a user's taste in music and film. We'll only consider one music genre, namely jazz (which we'll denote by `J`), and four films, "Waking Life" (denoted by `W`), "Borat" (denoted by `B`), "Cinema Paradiso" (denoted by `C`) and "Requiem for a Dream" (denoted by `R`). We'll assume that conditioned on whether the user likes jazz, the movie likes/dislikes are independent. The prior probability of liking jazz is 30%. We've defined the following (combined) conditional probability table, where "=1" means "likes". We are conditioning on the first column.

`J=1`	`W=1`	`B=1`	`C=1`	`R=1`
T	80	20	70	50
F	30	50	30	40

   a. What is the probability the user likes jazz, given that she likes the first and fourth movies but dislikes the second and third?
   b. How about the probability that the user likes jazz, given that she likes all the movies?

Part II: HMM naive solution (10 points)

In the remainder of this assignment, you will implement a basic Hidden Markov model. We'll use the HMM from our in-class part-of-speech tagging example, whose states are `PropNoun, Noun, Verb, Det`. The transition probabilities are the same as in the example shown in class. The observation probabilities are defined as follows, and are defined in the provided file `hmm_starter.m`.

State/Observation	john	mary	cat	saw	ate	a	the
`PropNoun`	0.40	0.40	0.10	0.01	0.05	0.03	0.01
`Noun`	0.25	0.05	0.30	0.25	0.05	0.05	0.05
`Verb`	0.04	0.05	0.04	0.45	0.40	0.01	0.01
`Det`	0.01	0.01	0.01	0.01	0.01	0.45	0.50

In a function `naive_solution.m`, write code to compute the probability of observing each of the following sentences, using the naive solution. You can map each word to a number that is its index into our vocabulary (the union of the column headers above, except the first one), then a sentence is just a vector of numbers; see Part IV for an example. Use the provided `permn.zip` to compute combinations with replacement, to get your list of possible state sequences.

Inputs:

- the transition matrix `A` (from `hmm_starter.m`),
- the observation matrix `B` (from `hmm_starter.m`),
- `N`, the number of states,
- `M`, the number of words in the vocabulary, and
- a vector of integers `sent` representing the sentence whose observation probability we want to compute.

Outputs:

- `prob`, the probability of observing the input sentence.

Part III: Testing HMM on part-of-speech tagging (10 points)

Finally, in a script `hmm_demo.m`, pick five of the sentences below, and compute their probability of occurrence. In a file `report.pdf/docx`, discuss what you observe about which of them seem more likely than others, and whether what you observe makes sense.

- "john saw the cat." (or using our mapping to numbers, `sent = [1 4 7 3];`)
- "john ate."
- "john saw mary."
- "mary saw john."
- "cat saw the john."
- "john saw the saw."
- "john ate the cat."

Submission: Please include the following files:

- `report.pdf/docx`
- `naive_solution.m`
- `hmm_demo.m`