Inference and Representation, Fall 2018

Problem Set 1: Bayesian networks

Due: Tuesday, September 11, 2018 at 12:59pm (as a PDF file uploaded to NYU Classes)

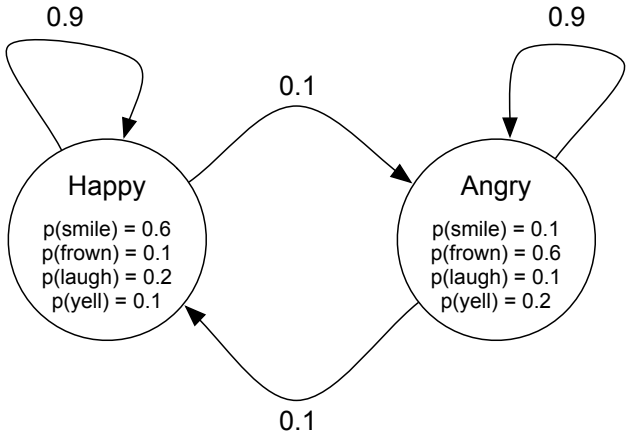
Important: See problem set policy on the course web site.

1. **Hidden Markov models**. Harry lives a simple life. Some days he is Angry and some days he is Happy. But he hides his emotional state, and so all we can observe is whether he smiles, frowns, laughs, or yells. Harry's best friend is utterly confused about whether Harry is actually happy or angry and decides to model his emotional state using a hidden Markov model.

   Let $X_d \in \{\text{Happy, Angry}\}$ denote Harry's emotional state on day d, and let $Y_d \in \{\text{smile, frown, laugh, yell}\}$ denote the observation made about Harry on day d. **Assume that on day 1 Harry is in the Happy state**, i.e. $X_1 = \text{Happy}$. Furthermore, assume that Harry transitions between states exactly once per day (staying in the same state is an option) according to the following distribution: $p(X_{d+1} = \text{Happy} \mid X_d = \text{Angry}) = 0.1$, $p(X_{d+1} = \text{Angry} \mid X_d = \text{Happy}) = 0.1$, $p(X_{d+1} = \text{Angry} \mid X_d = \text{Angry}) = 0.9$, and $p(X_{d+1} = \text{Happy} \mid X_d = \text{Happy}) = 0.9$.

   The observation distribution for Harry's Happy state is given by $p(Y_d = \text{smile} \mid X_d = \text{Happy}) = 0.6, p(Y_d = \text{frown} \mid X_d = \text{Happy}) = 0.1, p(Y_d = \text{laugh} \mid X_d = \text{Happy}) = 0.2$, and $p(Y_d = \text{yell} \mid X_d = \text{Happy}) = 0.1$. The observation distribution for Harry's Angry state is $p(Y_d = \text{smile} \mid X_d = \text{Angry}) = 0.1, p(Y_d = \text{frown} \mid X_d = \text{Angry}) = 0.6, p(Y_d = \text{laugh} \mid X_d = \text{Angry}) = 0.1$, and $p(Y_d = \text{yell} \mid X_d = \text{Angry}) = 0.2$.

   All of this is summarized in the following figure:



   Be sure to show all of your work for the below questions. Note, the goal of this question is to get you to start thinking deeply about probabilistic inference. Thus, although you could look at Chapter 17 for an overview of HMMs, try to solve this question based on first principles (also: no programming needed!).

   (a) What is $p(X_2 = \text{Happy})$?

(b) What is $p(Y_2 = \text{frown})$?

(c) What is $p(X_2 = \text{Happy} \mid Y_2 = \text{frown})$?

(d) What is $p(Y_{80} = \text{yell})$?

(e) Assume that $Y_1 = Y_2 = Y_3 = Y_4 = Y_5 = \text{frown}$. What is the most likely sequence of the states? That is, compute the MAP assignment $\arg\max_{x_1,\ldots,x_5} p(X_1 = x_1, \ldots, X_5 = x_5 \mid Y_1 = Y_2 = Y_3 = Y_4 = Y_5 = \text{frown})$.

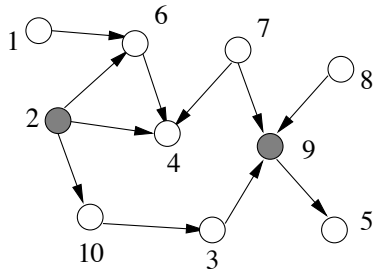
2. **Bayesian networks must be acyclic.** Suppose we have a graph $\mathcal{G} = (V, E)$ and discrete random variables $X_1, \ldots, X_n$, and define

$$f(x_1, \ldots, x_n) = \prod_{v \in V} f_v(x_v | x_{pa(v)}),$$

where $pa(v)$ refers to the parents of variable X_v in $\mathcal{G}$ and $f_v(x_v \mid x_{pa(v)})$ specifies a distribution over X_v for every assignment to X_v's parents, i.e. $0 \leq f_v(x_v \mid x_{pa(v)}) \leq 1$ for all $x_v \in \text{Vals}(X_v)$ and $\sum_{x_v \in \text{Vals}(X_v)} f_v(x_v \mid x_{pa(v)}) = 1$. Recall that this is precisely the definition of the joint probability distribution associated with the Bayesian network $\mathcal{G}$, where the f_v are the conditional probability distributions.

Show that if $\mathcal{G}$ has a directed cycle, f may no longer define a valid probability distribution. In particular, give an example of a cyclic graph $\mathcal{G}$ and distributions f_v such that $\sum_{x_1,\ldots,x_n} f(x_1, \ldots, x_n) \neq 1$. (A valid probability distribution must be non-negative and sum to one.) This is why Bayesian networks must be defined on *acyclic* graphs.

3. **D-separation.** Consider the Bayesian network shown in the below figure:



(a) For what pairs (i, j) does the statement $X_i \perp X_j$ hold? (Do not assume any conditioning in this part.)

(b) Suppose that we condition on $\{X_2, X_9\}$, shown shaded in the graph. What is the largest set A for which the statement $X_1 \perp X_A \mid \{X_2, X_9\}$ holds? The Bayes ball algorithm for d-separation may be helpful.

4. Consider the following distribution over 3 binary variables X, Y, Z:

$$\Pr(x, y, z) = \begin{cases} 1/12 & x \oplus y \oplus z = 0 \\ 1/6 & x \oplus y \oplus z = 1 \end{cases}$$

where $\oplus$ denotes the XOR function. Show that there is no directed acyclic graph G such that $I_{d-sep}(G) = I(\Pr)$.

5. **Naive Bayes classifier**

   (a) The dataset we will be using is a subset of 2005 TREC Public Spam Corpus, containing 9000 training examples and 1000 test examples. You can download it here

   `https://courses.cs.washington.edu/courses/csep546/12sp/psetwww/3/NDA.htm`

   Each line in the train/test files represents a single email with the following space-delimited properties: the first is the email ID (in the form /xxx/yyy), the second is whether it is 'spam' or 'ham' (non-spam), and the rest are words followed by their occurrence numbers. (Note that numbers may be words, so don't worry if a line contains multiple numbers in a row). The data has been pre-processed to remove non-word characters (e.g. '!') and to select features similar to what Mehran Sahami did in his original paper

   `http://robotics.stanford.edu/users/sahami/papers-dir/spam.pdf`

   though with larger cut-offs since our corpus is larger.

   (b) Using the training data, compute the prior probabilities $P(spam)$ and $P(ham)$. What is $P(spam)$?

   (c) Determine the vocabulary and compute the conditional probabilities $P(w_i|spam)$ and $P(w_i|ham)$.

   In this context, it is possible that none of emails labeled as say spam contain a particular word w_j. Then $P(w_j|spam) = 0$ and $P(spam)\prod_i P(w_i|spam) = 0$. To address this use a so-called m-estimate given by $p(w_j|spam) = \frac{n_c+mp}{n+m}$ where

   - n is the number of training examples which are spam
   - n_c is the number of examples, which are spam and which contain w_j
   - p is prior estimate for $P(w_i|spam)$, e.g., $p = 1/|Vocabulary|$)
   - m is weight given to the prior, i.e., number of deemed training examples, e.g., $m = |Vocabulary|$.

   In this context we consider each word as a training example, so n is the total number of words (in either ham or spam documents) and n_c is the number of times w_j appeared in those documents (including multiple occurrences in the same email).

   What are the 5 most likely words given that a document is spam? What are the 5 most likely words given that a document is ham?

   (d) Use these probabilities to classify the test data and report the accuracy (i.e. the percentage of correct classifications). Note that directly computing $P(spam|w_1, \ldots, w_n)$ and $P(ham|w_1, \ldots, w_n)$ can cause numerical precision issues, since the unnormalized probabilities are very small (i.e. the numerator in Bayes' theorem). Instead, you should compare the log-probabilities of being ham/spam.

   (e) Vary the m parameter, using $m = |Vocabulary| \times [1, 10, 100, 1000, 10000]$ and plot the accuracies vs. m. What assumptions are we making when the value of m is very large vs. very small? How does this affect the test accuracy?

   (f) If you were a spammer, how would you modify your emails to beat the classifiers we have learned above?

With your answers, please include your Python code and a report containing:

- A high-level description on how your code works.

- The accuracies you obtain.
- If all your accuracies are low, tell us what you have tried to improve and what you suspect is failing.