CS274b Homework #1

Learning in Graphical Models: Spring 2016

Due: Friday, April 15, 2016

Write neatly (or type) and show all your work!

Please remember to turn in at most two documents, one with any handwritten solutions, and one PDF file with any electronic solutions.

Problem 0: Piazza

Please join the Piazza forum at `http://piazza.com/uci/spring2016/cs274b`.

Problem 1: Bayes Nets

Suppose we have four random variables, W, X, Y, Z. Draw the (minimal) directed graphical model (Bayes' net) consistent with each of the following factorizations of the joint distribution. For each example, also give the number of (free) parameters required to specify the distribution (and why).

(a) $p(X)\ p(Y|X)\ p(Z|X,Y)\ p(W|X,Y,Z)$

(b) $p(W)\ p(X)\ p(Y)\ p(Z)$

(c) $p(Y)\ p(Z\,|\,Y)\ p(X|Y)\ p(W|Y)$

(d) $p(Z\,|\,X,Y)\ p(X)\ p(Y)\ p(W|X)$

(e) $p(W\,|\,X)\ p(X\,|\,Y)\ p(Y\,|\,Z)\ p(Z)$

(f) $p(W\,|\,X)\ p(Y\,|\,X)\ p(Z\,|\,Y)\ p(X)$

Problem 2: Conditional Independence (adapted from P&M, 6.2)

Consider the belief network shown in Figure 1, which describes how a number of variables related to the power in a building, and the activity of its occupants, are related. Answer the following questions about how knowledge of the values of some variables would affect the probability of another variable.

(a) Can knowledge of the value of *projector_plugged_in* affect your belief in the value of *Sam_reading_book*? Explain.

(b) Can knowledge of *screen_lit_up* affect your belief in *Sam_reading_book*? Explain.

(c) Can knowledge of the value of *projector_plugged_in* affect your belief in the value of *Sam_reading_book* given that you have observed a value for *screen_lit_up*? Explain.

(d) Which variables could have their probabilities changed if just *lamp_works* was observed?

(e) Which variables could have their probabilities changed if just *power_in_projector* was observed?

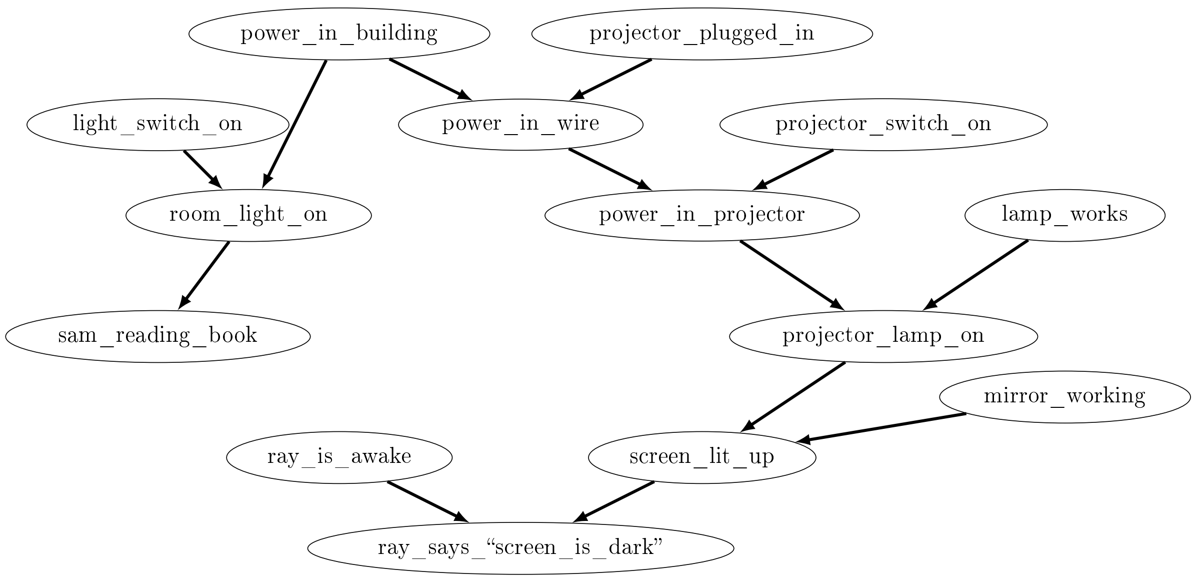

Figure 1: Bayesian network for the building electrical and projector model (P&M 6.21)

Problem 3: Partition functions

Consider a simple model over two binary valued variables $x \in \{0,1\}$ and $y \in \{0,1\}$:

$$p(x, y\,;\,\theta) = \exp\left[\theta_x x + \theta_{xy}\, x\, y - A(\theta_x, \theta_{xy})\right],$$

so that $A(\theta)$ normalizes the distribution.

(a) Compute / write the log partition function $A(\theta)$.

(b) For fixed $\theta_{xy} = 1$, sketch / plot $A(\theta)$ as a function of θ_x; you should verify (just visually) that it is (appears) convex.

(c) Compute the gradient $\nabla A(\theta)$ at $\theta = [\theta_x,\ \theta_{xy}] = [1, 2]$. Interpret its value.

Problem 4: Learning Markov models (Matlab)

In this problem, you will learn a simple Markov model (a hidden Markov model structure) for predicting protein secondary structure. Download the data set associated with the homework, and unzip it (it will create a large number of text files); each of these is a labelled secondary structure and a protein sequence. (These data come from Astral, `http://astral.berkeley.edu/`; see that page for more information.)

I suggest that you use Python+NumPy to manipulate the data and resulting matrices, although you may use any software you like. Please include your code with your submission.

First, load the data and convert it from text strings into numeric indices (to make things easier):

```
import numpy as np
from os import walk
mypath = 'proteins/'  # use path to data files
_, _, filenames = next(walk(mypath), (None, None, []))

mSeq = len(filenames)        # read in each sequence
```

```
o,x = [],[]
for i in range(mSeq):
    f = open( filenames[i] , 'r')
    o.append( f.readline()[:-1] )  # strip trailing '\n'
    x.append( f.readline()[:-1] )
    f.close()

xvals, ovals = set(),set()  # extract the symbols used in x and o
for i in range(mSeq):
    xvals |= set(x[i])
    ovals |= set(o[i])
xvals = list( np.sort( list(xvals) ) )
ovals = list( np.sort( list(ovals) ) )
dx,do = len(xvals),len(ovals)

for i in range(mSeq):       # and convert to numeric indices
    x[i] = np.array([xvals.index(s) for s in x[i]])
    o[i] = np.array([ovals.index(s) for s in o[i]])
```

Each $x^{(i)} =$ `x[i]` is one observed sequence, each with different lengths.

(a) From your data, estimate $p(x_0)$, the initial state distribution for a given sequence.

(b) Now, estimate the transition probability matrix, $T_{ij} = \Pr[x_t = j | x_{t-1} = i]$. What is the stationary distribution of this model? (Print T_{ij} for i, j in the first five state values.)

Note that in this representation, if we represent $p(X_0)$ as a row-vector v, then

$$p(X_1 = j) = \sum_i \Pr[X_t = j | X_{t-1} = i] p(X_0 = i) = v \cdot T.$$

(c) Estimate the emission probability matrix, $O_{ik} = \Pr[o_t = k | x_t = i]$. (Again, print O_{ik} for i, k in the first five values.)

(d) Notice that our model is not a fully generative model: it does not explain the sequence length (it takes it as fixed and known). How might we modify the model (including training, etc.) to also explain sequence length? (You do not need to actually do this; just explain.)

(e) Complete the code to compute the marginal probability of the state x_t given the observation sequence $O = [o_1, \ldots, o_L]$, for each t:

```
function markovMarginals(x,o,p0,Tr,Ob):
  '''Compute p(o) and the marginal probabilities p(x_t|o) for a Markov model
     defined by P[xt=j|xt-1=i] = Tr(i,j) and P[ot=k|xt=i] = Ob(i,k) as numpy matrices'''
  dx,do = Ob.shape()   # if a numpy matrix
  L = len(o)
  f = np.zeros((L,dx))
  r = np.zeros((L,dx))
  p = np.zeros((L,dx))

  f[0,:] = ...   # compute initial forward message
  log_pO = ...   # update probability of sequence so far
  f[0,:] /= f[0,:].sum()  # normalize (to match definition of f)
  for t in range(1,L):    # compute forward messages
    f[t,:] = ...
    log_pO += ...
    f[t,:] /= f[t,:].sum()

  r[L,:] = 1.0  # initialize reverse messages
  p[L,:] = ...  # and marginals

  for t in range(L-1,-1,-1):
    r[t,:] = ...
    r[t,:] /= r[t,:].sum()
    p[t,:] = ...
    p[t,:] /= p[t,:].sum()

  return log_pO, p
```

I suggest you verify your code by hand on a small model and simple observation sequences to test it. Please report how you tested your code for correctness.

To verify your code for me, please also report the marginal probability distributions (as vectors) for $p(x_6^{(0)}|o^{(0)})$ and $p(x_9^{(2)}|o^{(2)})$, along with the log-probability of the observation vector, $\log p(o^{(4)})$.

(f) **For enrichment; not graded** Write code to compute the most likely state sequence for x.