CS B553 Assignment 4: Markov Chains and Learning

Spring 2013

Due: Sunday April 28, 11:59PM. (You may submit up to 48 hours late for a 10% penalty.)

This assignment will give you some experience with Markov Chains and learning techniques. As with past assignments, you may work on this assignment alone or in a partnership with another student in the class. If you work with a partner, please submit only one copy of your submissions for grading.

This assignment consists of two questions; the first one does not (necessarily) have a programming component, while the second one does.

1. **Markov chains.** The rental car agency of a hypothetical mid-western airport has N cars available to rent, arranged in a long line right outside the baggage claim area. Each car has a unique ID number, ranging from 1 to N painted on top of the car. The ID number is also a measure of quality, so that airport visitors generally want the cars with higher ID number; in fact, it turns out that the probability of a given visitor choosing the car with ID i is,

$$P(i) = \frac{i}{\sum_{j=1}^{N} j}.$$

When the airport first opened, the cars were arranged in order from 1 to N, with car 1 closest to the baggage claim area and N furthest away. The airport is not very heavily traveled, so that exactly one car is rented per day, and it is always returned before the end of the day. When a traveler rents a car, he or she chooses it according to $P(i)$ above, and then returns it to the parking spot closest to the baggage area. When he or she returns it, it's parked at the front of the line, i.e. right beside the baggage claim area, and the rest of the cars are pushed forward one spot until all spots are full. The rental car agency would like to reason about the ordering of the cars at various points in time.

This system can be modeled as a Markov Chain, with a set of states $Q \subseteq \mathcal{Z}^N$. For instance, if $N = 3$, then there are 6 states, where for example $(1, 2, 3)$ corresponds to the car 1 being closest to the baggage claim, then 2, then 3, while state $(2, 1, 3)$ means that car 2 is closest to the baggage claim and 1 and 3 are parked behind.

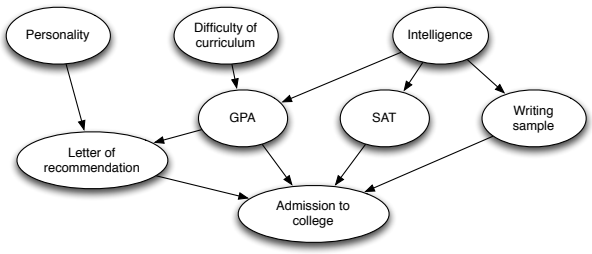
   (a) Write down the Markov Chain, including the set of states, transition probabilities, and initial distribution, for the case of $N = 3$.

   (b) Prove that the Markov Chain is ergodic and hence a stationary distribution exists.

   (c) What is the stationary distribution in the case of $N = 3$? *(You can figure this out by hand, write a program, or use mathematical software like Matlab or Mathematica. In any case, please submit your work – your hand calculations, your program, or your interactive Matlab session.)*

   (d) What is the stationary distribution for N in general?

2. **Learning.** Consider the following Bayes net, which predicts whether a given student will be admitted to college based on various factors:



Each of these variables is binary valued, e.g. bad or good Personality (P), low or high Difficulty (D), low or high Intelligence (I), low or high GPA (G), low or high SAT score (S), bad or good Writing sample (W), weak or strong Letter of recommendation (L), and successful or unsuccessful Application to college (A).

Your goal is to learn the conditional probability distributions for this Bayes net, from training data, using maximum likelihood estimation. The complication is that the training data is incomplete: I is never observed, because it's difficult (maybe impossible) to measure intelligence directly. Meanwhile other variables may be missing because some college applications are incomplete.

More specifically, suppose you're given a dataset consisting of a set of training exemplars, $D = \{Z_1, Z_2, ..., Z_L\}$. Recall that if a given variable X_i and all of its parents $\text{Pa}(X_i) = \{Y_1, ...Y_m\}$ are always observable, then one can simply use maximum likelihood estimation: each entry in the corresponding conditional probability table should be set equal to,

$$P(X_i = x_i | Y_1 = y_1, ...Y_m = y_m) = \frac{M[X_i = x_i, Y_1 = y_1, ...Y_m = y_m]}{M[Y_1 = y_1, ...Y_m = y_m]}$$

where $M[\cdot]$ counts the number of exemplars in D that satisfy the given conditions.

For the variables that are not observable, we'll have to use EM. Start by setting the conditional probability distributions (other than ones estimated above) to some initial values, $\theta^{t=0} = (\theta_1^{t=0}, ..., \theta_N^{t=0})$, where N is the number of variables in the network and $\theta_i^{t=0}$ represents the estimated conditional probability distribution table encoding $P(X_i|\text{Pa}(X_i))$. Now iterate back and forth between the following two steps:

- **E-step:** For every training exemplar Z_j, and for every latent variable X_i, compute the marginal distribution $P(X_i|Z_j, \theta_i^t)$. In other words, for every variable that you can't observe, estimate the probability that it has value 0 or a 1 given the variables that you can observe, and the current estimated probability distributions θ^t.
- **M-step:** Now compute maximum likelihood estimates using the marginal distributions computed during the E-step, to estimate a new model θ^{t+1}. This new model will have conditional probability entries computed as,

  $$P(X_i = x_i | Y_1 = y_1, ...Y_m = y_m) = \frac{\bar{M}[X_i = x_i, Y_1 = y_1, ...Y_m = y_m]}{\bar{M}[Y_1 = y_1, ...Y_m = y_m]},$$

  very similar to the fully-observable case above, except that the $\bar{M}[\cdot]$ are sums of probabilities across the exemplars instead of counts. For example,

  $$\bar{M}[Y_1 = y_1, ...Y_m = y_m] = \sum_{j=1}^{L} P(Y_1 = y_1, ..., Y_m = y_m | Z_j, \theta^t).$$

The E-step and M-step are run iteratively until convergence.

Here's what to do in this assignment:

(a) Write a program that implements EM to estimate these parameters. We've given you some training data on OnCourse. Each line of the file is an exemplar, and has the following format:

```
D I P G L S W A
```

where each variable value is `0` for low/bad, `1` for high/good, or `x` for missing. You may use any language of your choice for this. Note that the E-step involves doing inference on the model. You can use the inference algorithm of your choice here. Variable elimination or sum-product are good choices, for instance, or you could use Gibb's sampling for some variety. You may also be able to do brute-force inference, since there is a small number of binary-valued variables. Just explain your choice in your report.

(b) Once you have (a) working, test in using both (i) random initialization of parameters at $t = 0$ and (ii) initialization of parameters set by your own intuition. Do you get different answers? Please show the learned conditional probability distribution tables in your report.

(c) Test your parameters on the test dataset provided on OnCourse. For each exemplar, take all variables except for A as observed, compute the most likely value for A, and then compare to the actual value of A. For what percentage of the exemplars does your trained net classify A correctly?

Important hints and warnings

Design decisions. You'll likely encounter some decisions as you write your answers that are not specified in this assignment. Feel free to make some reasonable assumptions in these cases, documenting them in your report, or to ask on the OnCourse forum for clarification.

Academic integrity. The goal of this assignment is to help you learn about Markov chains and learning. We thus require that you (and your partner) do the work on this assignment yourselves. You and your partner may discuss the assignment with other people at a high level, e.g. general strategies to solve the problem, language syntax, etc. You may also consult printed and/or online references, but you must cite these materials in the documentation of your source code and explicitly state which parts of your code were influenced by which sources. However, the solutions and code that you (and your partner) submit must be your own work, which you personally designed and wrote. You may not share written solutions or code with any other students except your partner, nor may you possess code or solutions written by another person who is not your partner, either in whole or in part, regardless of format. Please note that copying or modifying solutions or code from a student, from online sources, or from a textbook is a violation of the academic integrity policy. As described in the course syllabus and the IU Academic Handbook, University policy requires us to investigate suspected cases of academic integrity, to assign penalties, and to report to the Dean of Students for possible additional sanctions (up to and including expulsion from the University).

That said, we want you to succeed and are very happy to help if you have questions about the assignment. Stop by office hours, write a question on the OnCourse forum, or set up an appointment to meet with us.

What to turn in

Turn in two files, via OnCourse:

1. A .zip archive file containing your source code.
2. A separate text file or PDF with a report about your work and the written answers requested above. Explain, at a high level, how you implemented it, what design decisions and other assumptions you made, any problems you had to overcome, etc. Give credit to any source of assistance (students with whom you discussed your assignments, instructors, books, online sources, etc.). Also give the accuracy details requested above (on the results of your testing on our sample datasets).