CSC 535 – Probabilistic Graphical Models

Assignment Six

Due: 11:59pm (*) Wednesday, Oct 31.

(*) There is grace until 8am the next morning, as the instructor will not grade assignment before then. However, once the instructor starts grading assignments, no more assignments will be accepted.

Weight is about 5 points

This assignment and the next can be done in groups of 2-3. You should create one PDF for the entire group, but everyone should hand in a copy of the PDF to help me keep track. Make it clear on the first page of your PDF who your group is. Group reports are expected to be higher quality than individual ones.

The purpose of this assignment is to implement a Bayes nets with a more complex specification than we have done so far. This assignment is designed to set us up for the next assignment on inference. Some of the code developed, or results found here might be needed for a later assignment also. KEEP UP! (And if you are having trouble doing so, let the instructor know).

Deliverables

Deliverables are specified below in more detail. For the high level perspective, you are to provide a program to output a few numbers and to create figures. You also need to create a PDF document that tells the story of the assignment including output, plots, and images that are displayed when the program runs. Even if the question does not explicitly remind you to put the resulting image into the PDF, if it is flagged with **($)**, you should do so. The instructor should not need to run the program to verify that you attempted the question. See

http://kobus.ca/teaching/grad-assignment-instructions.pdf

for more details about preparing write-ups. While it takes work, it is well worth getting better (and more efficient) at this. A substantive part of each assignment grade is reserved for exposition.

Sex in flatland

The sex of the magnetic-quad-limb (MQL) creatures in flatland is hard to tell. Let's write a computer program to help them out. There are twice as many males as females. The bodies of males are circular and that of the females are hexagonal. These shapes are easily confused. In particular, the circular male body has a 30% chance of being detected as being a hexagon, and the hexagonal female body has a 30% chance of being detected as a circle.

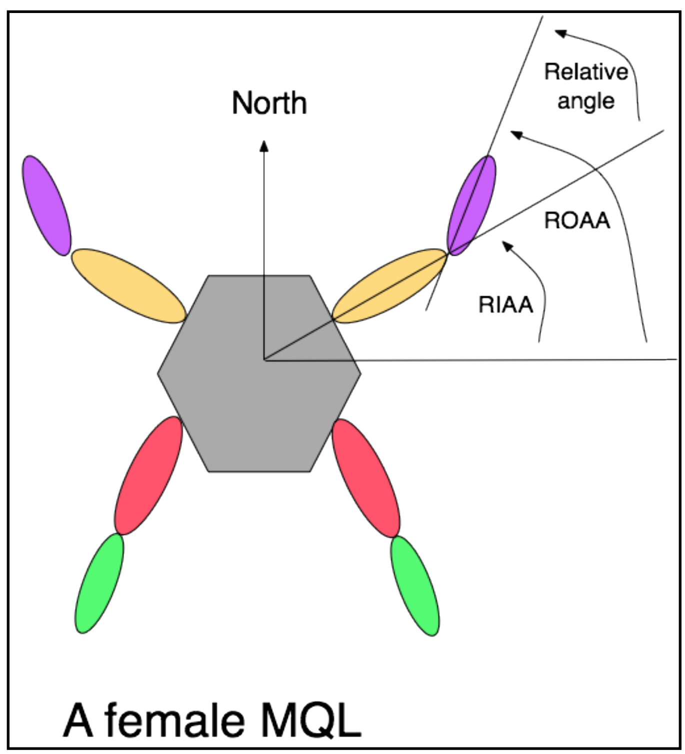
The color of the bodies are gray. The bodies of the MQL creatures always align with the external (known) magnetic field. This means that there is a well defined coordinate system for the angles of each limb leaving their bodies. MQLs have two arms and two legs, each having an inner segment attached to the body, and an outer segment attached to the inner segment. The inner arm segments (IAS) are golden, the outer arm segments (OAS) are purple, the inner leg segments (ILS) are red, and the outer leg segments (OLS) are green.

The colors can be reliably detected, and it is always clear which of the identically colored blobs (e.g., green blob) are the left one of the pair and the right one of the pair (e.g., for green, we know left OLS (LOLS) and which is the right OLS (ROLS)). Notice however, that the colors do not tell you anything about the sex of the MQL.

Other than the (unreliable) body detection, evidence for the sex of the MQLs comes from the limb angles. However, because the MQLs are quite furry, angles are also unreliably measured. MQLs are symmetric, so the right inner arm angle (RIAA) and left inner arm angle (LIAA) sum to 180 degrees. Similarly, the right inner leg angle (RILA) and left inner leg angle (LILA) sum to 540 degrees. Further, RIAA is between 0 and 90 degrees, and RILA is between 270 and 360 degrees. Note that RIAA and RILA determine the LIAA and LILA respectively. The distributions of RIAA and RILA are different for males and females as specified below. Finally, the outer limb angles, measured with respect to the inner limb angles (see figure) are independent of the inner limb angles. The distributions are the same for arm outer segments and leg outer segments, and they are the same for left and right. This distribution does not depend on sex.

We will consider that angles are discretized into 9 values, with the value 5 corresponding to the middle of the relevant quadrant. Angles will step in 15 degree increments, although this is only relevant for visualization. To specify distributions, we will provide 5 numbers which correspond to values 3 through 7, with the other values being zero. We need a bit extra so that when an inner and outer angles add up, the result is in our range.

It is best to think of sampling one angle (outer), conditioned on another (inner), with the distribution of the outer angle having its center placed over the sampled inner angle. So, for example, the largest possible RIAA is bin 7. The largest possible ROAA is two bins to the right of the center of its relative distribution. So, the absolute angle is two hops to the right of bin 7, which is bin 9. So, we should never sample an angle in bin 10. This is



for the model. The observables will also not be outside the range, based on rigging the detector (see below).

One additional point about this representation is that flipping between left and right angle values where there is symmetry requires flipping the angle value across the middle of the distribution. For example, right and left angles 3 and 7 correspond. In addition, if you are studying the relation between angle values and the visualization, the semantics of arm numbers and leg numbers are rotated by 90 degrees. For example, an angle value of 8 is a vertical right arm, but it is a horizontal right leg.

The angular distributions for bins 3 through 7 are:

- Females have RIAA distributed in proportion to: 0,0,10,30,50
- Males have RIAA distributed in proportion to: 10, 20, 30, 20,10.
- Females have RILA distributed in proportion to 10,20,30,20,10.
- Males have RILA distributed in proportion to: 50, 30, 10, 0, 0.

Both males and females have both outer segments distributed around the angle of the inner limb that they are attached to. The relative angles are distributed in proportion to: 30,20,10,20,30 (yes, the are bent).

Observed data. For each true angular value, the angle that is actually measured has 20% chance of being measured in bin numbered one less (if there is one), 10% chance of being measured as coming from the bin numbered two less (if there is one), and similarly, it has a 20% chance as appearing to come from bin numbered one more (if there is one) and 10% chance of appearing to come from the bin numbered two more (if there is one). This measurement error is independent for each measurement made. Hence, although we know that MQLs are symmetric, they may not appear to be. We will assume that bins outside 1 to 9 **never occur**. Specifically, you can assume that the detector only works for these angles and angles outside this range are read as being from the closest boundary angle.

The model for the outer segments are defined in terms of the relative angle, but what is measured is the absolute angle (like a compass bearing). It is also important not to confuse the difference of the *measured* inner and outer angles with the true relative angle—the former does not have the statistics specified above due to the measurement process. For example, you can *observe* relative angles that would seem to be impossible if you were to assume no measurement error.

We will assume that we never see anything that is impossible, which is a safe assumption because we will work with synthetic data. (When you get to the next part of the assignment, you might choose to create some small examples for testing your code, but then make sure that your test case is also possible with respect to the above).

1. Produce the 9x9 table *p(measured_angle | angle)* with the rows of the table being for different (model) angles and the columns being for different measured angles ($). Do you think that the variance of the measurements will be the same for each angle ($)?

2. Provide a graphical model for the hidden variables and the observable variables **($)**.

   Hint. It is probably easiest to create a hidden variable for symmetric pairs of limbs, and use those to "generate" observations which then happens a bit differently on the left and right sides. Enforcing symmetry without some kind of a top-down mechanism (such as this one) is not typically possible in a directed model.

3. Provide a probabilistic formula for the joint distribution of hidden variables and the observable variables **($)**.

4. (+) Write a program to generate samples from the probability distribution, and draw pictures indicating generated male and female MQLs. (Matlab[1] and Python[2] code is available to help generate images for those who do not want to do their own graphics). Generate some samples, and provide 10 of them **($)** in two representations: 1) The idealized individual; and 2) The observed appearance is (i.e., (1) with sampled measurement noise).

5. Can you tell them apart (i.e., identify male versus female) reliably by just looking at the figures from the previous question using (1)? How about (2)? **($)**.

This assignment has one recommended optional questions that will be considered for extra credit. Even if you do not have time to write it up, it is worthwhile to think about it.

6. (**) Based on question 4 (part 2), you can generate synthetic data to experiment with inference. Specifically, you could consider how well an algorithm infers the MQL from observations, by generating (MQL_{MODEL}, MQL_{obs}) and seeing how well you can infer MQL_{MODEL} from MQL_{obs}. To begin, consider critically the problem we have set up for ourselves. Specifically, we have a model we trust and MQL_{obs}. Perhaps we interpret our task at inferring the MQL_{MODEL}. Is this a good interpretation? Can you think of others? Assuming this interpretation, and assuming that we can compute the MQL_{BEST} = ArgMax p(MQL| MQL_{obs})}, have we succeeded? IE, is it true that MQL_{MODEL} =MQL_{BEST}?

   In the next assignment, we will implement inference along these lines using the sum-product and max-sum algorithms. However, to better understand sampling, consider the following algorithm for doing so (input is MQL_{obs}).

   Loop

      Generate MQL (see below)

      Compute p(MQL, MQL_{obs}), and if it is bigger than the current max, bump the current max to this value, and save MQL as MQL_{BEST_EST}

   End loop (when laptop battery is low)

   Output MQL_{BEST_EST}

   To generate MQL, consider 1) Uniformly sampling each parameter, 2) Sampling as you did in question 4; and 3) Sampling p(MQL | MQL_{obs}) (this is actually not trivial, but assume that you are provided this capability). Can you provide any estimate of how many iterations (on average) you will need to find the true best value MQL_{BEST} (highest possible MQL_{BEST_EST})? If needed, you can assume that an oracle knows MQL_{BEST} and is willing to tell you for the purpose of the assignment.

[1] http://kobus.ca/teaching/cs535/code/draw_mql.m
[2] http://kobus.ca/teaching/cs535/code/draw_mql.py

What to Hand In

Hand in the PDF file hw6.pdf with the story of your efforts into D2L. If you wrote code for this assignment (you should!), hand in a program hw6.<suffix> (e.g., hw6.m if you are working in Matlab). If you are working in groups, the code and the PDF can be the same for each person in the group, but each member should hand in a copy to help me keep track of things. PDFs from groups are expected to be of higher quality than individual ones.