CSC 535 – Probabilistic Graphical Models

Assignment One

Due: 11:59pm (*) Thursday, August 30.

(*) 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 about 4 points

This assignment should be done individually

The first purpose of this assignment is to sort out the software that you will use for this course. There is no requirement to use any particular language, or to use the same language for every assignment or even to use only one language to solve a particular problem. However, if you do not have any preference, I suggest Matlab, even if you do not know it. In a similar course that I have taught several times, most folks used Matlab exclusively, with many of them learning Matlab in the process. A few did at least some of the work in R, and I recall some trying Python and C/C++ (with IVILAB library support) for some parts. In short, previous incarnations of the course has worked well with Matlab. In addition, for better or worse, much machine learning is done in Matlab. Python might take over, but these things take time.

If you would like to consider using Matlab, and are not familiar with it, I strongly suggest working through some of the parts of the optional assignment zero. (If you have taken my computer vision class, you have done many of these questions already).

If you would like to consider other alternatives, you will need (A) matrix operations including inverting them and finding eigenvectors and eigenvalues; (B) plotting of data and simple functions; (C) displaying of images; and (D) basic programming with conditionals, loops, arrays, and evaluation of basic functions such as exp(), and log(). It is possible that you will want to switch to Matlab for certain parts of certain assignments, so be prepared for that.

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.

---

1. Random numbers

Set the random number generator seed to 0 and then produce 1000 throws of two (6-sided) die. Use the result to estimate the probability of double sixes. Report what you did and the result ($). Now run your code 9 more times, making sure that the random number generator is not reset. Report the results, and comment how many times you got the same estimate as the first time **($)**. Finally, set the seed to 0 a second time, and report whether you get the same result as the first time **($)**. Explain why it is often important to have random number sequences that are not really random, and can be controlled **($)**.

2. Random images

Download the text file http://kobus.ca/teaching/cs535/data/tiger.txt and read it in as a matrix. The number of rows in matrix should be the number of lines in the file, just as one would expect. Display the matrix as a grayscale image, and put the image into your report **($)**.

We will assume that grayscale means that each pixel has a brightness represented by 8 bits per pixel (256 shades of gray). This means that the grayscale tiger image could be thought of as a (uniform) random sample of an integer between 0 to 255, repeated for each pixel in the 236x364 grid. Create two images that are such random samples and put them into your report ($) [Three sub-figures side by side probably works best]. Are the new images recognizable as scenes in the world like the tiger image? **($)**. Create and display some additional examples as needed (but do not put them into your report) to comment on (A) the difference (if any) between the random examples and whether a different random seed would change your conclusions, and (B) the relationship (if any) between the generated random images and everyday visual content, and (C) does this experiment tell you anything about everyday visual content ($).

Can you estimate how many times on average you would have to sample grayscale images of size (236, 364) to get the exact tiger image? [Do either version A or B as follows, or both for extra credit] **($)**.

(A) The problem as most simply stated (as above) is a bit tricky and relies on more background then we want to assume at this point because the stream of images can have duplicates. For the purpose of this assignment (version (A)), you can instead imagine that the universe has a single copy of each of these images (like a deck of cards), and gives you one at a time. When all images have been handed out, it stops.

(B) (**) Do the problem as stated where the system keeps handing you images, and due to duplicates it is possible that you could have to wait an arbitrarily long period of time to get a particular one (but you will get it eventually).

3. A quick check that you can provide plots

(A) Provide a plot of both sin(x) and cos(x) over the domain - $[-\pi,\pi]$. Each curve should have a different color Put the figure into your PDF with an informative caption **($)**.

(B) Provide a bar plot for the histogram counts for the grey values of the image you used in the preceding question ($). Provide a similar plot where the counts have been converted to empirical probabilities by scaling them so that they sum to one ($). Put the plots into your PDF with an informative caption **($)**.

If you have not encountered histograms before, please look them up (e.g., on Wikipedia).

4. Integration

We often approximate continuous functions by sampling them at frequent intervals, and storing the values a vector. Derivatives and integrals can then be approximated by finite differences (http://en.wikipedia.org/wiki/Finite_difference) and Riemann sums (http://en.wikipedia.org/wiki/Riemann_sum), respectively. In this class, for example, we might use integrals to evaluate the probability of an event under a continuous probability distribution.

Assume the heights of adult men follow a normal distribution with mean 70 inches and standard deviation of two inches. Let's evaluate the probability of a man having a height between 68 and 80 inches.

The normal distribution function is defined as:

$$\mathcal{N}\left(x;\mu,\sigma^2\right)=\left(2\pi\sigma^2\right)^{-1/2}\exp\left(-\frac{\left(x-\mu\right)^2}{2\sigma^2}\right) \tag{1}$$

Here, σ is 2 inches, and μ is 70 inches. (The semi-colon in the above might confuse. It is relatively common to separate the last block of function parameters with a semi-colon when those parameters are being thought of as constants).

Begin by creating a vector of x values, evenly spaced over the range [68, 80] at increments of 2 inches. Next, compute a y vector, containing the values of $\mathcal{N}\left(x;\mu,\sigma^2\right)$. Plot the result as a bar chart together with the curve for the formula over a wide enough range that it close to zero at the extremities of the range **($)**. The bars are an approximation of the function in the region that we will be integrating. It is not smooth as we are discretizing a continuous function by sampling it at intervals of size 2. Decreasing the interval between x-values will make the function appear smoother, but will require more computing time to operate on.

Now compute the Riemann integral (http://en.wikipedia.org/wiki/Riemann_sum), using the y values as the rectangle heights, and the delta-x (2 inches) as the rectangle width. This is equivalent to summing the values of y and multiplying by delta-x. The result is an approximate integral which represents the probability of a man being between 68 and 80 inches. Report your estimate **($)**.

Now compute the integral over a much wider range of values (say [20,120]). State in your writeup what you expect the value to roughly be ($). Since you have a good idea what the result should be, you can now experiment with changing the value for delta-x and observe the change in the result. Try this for some values of delta-x that both larger, equal to, and smaller than the one just used. Plot the error of the estimate verses delta-x (or perhaps log(delta-x) if that is more informative). Make sure your cast a wide enough net on delta so that your results can tell a story **($)**.

5. Formalizing research questions

This problem is motivated by the anticipation of having you apply methods that we are learning to problems that you are actively working on. It is open ended and vague, perhaps too much so. For some, it might be hard to be specific at this time. ***But I would like to know what you care about that might be addressable by probabilistic graphical models****, and this question will also help get you prepare for later homework where you might extend your description to include a probabilistic graphical model for your domain. This question will be graded mostly on exposition. If you are too stuck, consider one of the optional problems below.*

Tell me about one or more problems that you find interesting. Pictures are likely helpful. This might be something you are actively working on for PhD or MS research, or a recent project, or problems from other classes, or something from a paper you have read, or simply something that you find interesting **($)**.

Develop notation for elements such as constants, parameters, and data in your problem. You should be clear what the data is. Be careful to provide indices where needed (e.g., perhaps you have data about temperature over time, for multiple objects. The data points then need both an object and a time index). You may need to specify what kind of quantities are being represented (continuous, categorical, ordinal), what the units are, and what the ranges are **($).**

Try to express what problem(s) you trying to solve mathematically, or at least with clearly defined input and output **($).**

> If you have done all the questions until here, then congratulations, you are done! However, if you are looking for alternatives, or extra problems, feel free to keep reading.

6. Random images (II) (**)

Continuing question 2, think about how long you would expect the generation of an image that is similar enough to the tiger image that a person would probably assume that the images are basically the same. You will have to make some reasonable but somewhat arbitrary assumptions here. There is no well defined answer here, but a good answer will state assumptions clearly, and in such a way that they could be tweaked and tested **($).**

Possibly helpful comment. Every answer I can think of for this problem can be related (at least with some imagination) to compression.

7. Random images (III) (**)

Continuing question 2, but now think about how long you would expect to wait for the random generation of an image that a human would declare is an image of something in the world, rather than the result of someone playing with a computer. Again, we are more interested in the analysis than the result (which is not known). Some creativity will be needed to approach this problem **($).**

What to Hand In

Hand in a program hw1.<suffix> (e.g., hw1.m if you are working in Matlab) and the PDF file hw1.pdf with the story of your efforts into D2L.