University of Washington Bothell
Computing & Software Systems

Course title: Introduction to Machine Learning
Course number: CSS 581
Term: Winter 2014
Instructor: Jeff Howbert

Exercises 3

Date assigned: Jan. 15, 2014
Date due: Jan. 20, 2014

Question 1) [2 points each] For each example below, state whether the attribute is i) discrete or continuous, and ii) nominal, ordinal, interval, or ratio. If you're doubtful about any part of an answer, add a few words of explanation. [Before doing this question, review Chap. 2.1.1 and slides 5-9 in slide set 03_data.pdf.]

a)   Brightness outdoors as measured by a light meter.
b)   Brightness outdoors as indicated by people's subjective judgments (e.g. "it's pretty bright", "hard to see").
c)   Angles as measured in degrees between 0 and 360.
d)   Telephone area code.
e)   Bronze, silver, and gold medals as awarded at the Olympics.
f)   Height above sea level in meters.
g)   Number of patients in a hospital.
h)   Ability to pass light in terms of the following values: opaque, translucent, transparent.
i)   Military rank.
j)   Density of a substance in grams per cubic centimeter.
k)   Coat check number. (When you attend an event, you often can give your coat to someone who, in turn, gives you a number to use to claim your coat when leaving.)

The remaining questions are designed to help you start coding in MATLAB. They use a mix of code samples and tips and tricks to guide you through a series of coding exercises. In several places, I expect you to figure out details of syntax by consulting MATLAB's Help. Since the material builds from one question to the next, it will be easiest to do them in order. For each question, I have tried to state clearly what you need to copy into the answers you turn in.

To start, open MATLAB, either from a shell command line (Linux), or by clicking on a desktop shortcut or Start menu entry (Windows). Then load the Fisher iris dataset with the command 'load fisheriris'.

First, some practice with indexing, and combining it with basic mathematical operations. When accessing matrices, the row(s) are specified by one index number (or vector of numbers), and the columns by another index number (or vector). For example, if `A` is a 3 x 5 matrix, you can pull out the elements in rows 2 and 3 and columns 1 and 4 with:

```
A( [ 2, 3 ], [ 1, 4 ] )
```

The result is a 2 x 2 matrix.

Each index vector can have 0, 1, or many integers in it, as long as they are legal index values. For indexing purposes, it doesn't matter if the vector is a row or column vector.

Recall from lecture that:
   i.   You can use colon syntax to create vectors containing a range of integers, e.g. [1 : 25].
   ii.   You can also use colon syntax to create a vector of integers with equal spacing between them, e.g. [1 : 3 : 28] gives a vector containing every third integer between 1 and 28.
   iii. If you specify just a colon, instead of a vector of integers, that selects all the rows or all the columns.
   iv.   If you take an operator that works on vectors and apply it on a matrix, the operator is applied separately to each *column* in the matrix, producing a vector of answers, one answer for each column in the matrix. [If this doesn't make sense, review the examples using mean() and std() in `matlab_demo_2.m`.]

If at any time you want to check your answers in the Command Window against the contents of an array, go to the Workspace Window and double-click on the icon next to the array name; this will open the Variable Editor.

Question 2) [2 points each] Perform the operations below on the `fisheriris meas` array by entering commands in the Command Window. For each operation, cut-and-paste from the Command Window into your answer document: i) the command you typed, and ii) the answer output by MATLAB.
   a) Find the sums of each of the four columns in the array. (Try searching on "sum" in the MATLAB Help system, or the command 'help sum'.)
   b) Find the sums of just columns 2 and 4.
   c) Pull out samples (rows) 27 through 48 and find the maximum values in each of the four columns of this set of samples.
   d) Pull out the odd-numbered samples (rows) between 3 and 33 and find the minimum values in just columns 1 and 2.

Random sampling is a very important core technique in machine learning. For predefined sampling, you can use the kinds of structured indexing illustrated in the previous question. To sample randomly,

however, you need to have a random number generator (RNG) running. Here's the standard way to start a RNG in MATLAB:

```
randSeed = 1;
stream = RandStream( 'mt19937ar', 'seed', randSeed );
RandStream.setGlobalStream( stream );
```

Look up "RandStream" in Help to learn more about the properties and capabilities of this object.

The `randSeed` argument allows you to start the RNG in a defined state. Being able to do this is critical for reproducing any work that involves random processes.

Once the RNG is running, you can invoke any command that needs random numbers, and it will automatically draw as many as it needs from the RNG.

Now let's learn the basics of writing and executing a script. From the top toolbar on the MATLAB GUI, you can open a new script in the Editor with `File > New > Script`. You can also open an existing script in the Editor by double-clicking on it. The Editor is actually a full development environment that supports debugging, etc. The text editing operations are pretty much like those in Word, e.g. highlighting to cut and paste. Explore the toolbar at the top to get a feeling for the Editor's features.

One quick tip: to comment a line of code, place the cursor on it and hit Ctrl-R. To comment a block of code, highlight it first, then Ctrl-R. Ctrl-T uncomments a line or block. You will find quick commenting and uncommenting to be very useful when you start assembling unfamiliar syntax into new code modules.

From inside a running script, there are two basic ways to display the value of a variable or the result of a calculation or assignment from inside a running script:
   i.   Leave the semicolon off the end of the command.
   ii.   Use the `disp()` command. For example, `disp( a )` displays all the values in vector `a`, and `disp( max( a ) )` displays the value of the largest element in `a`.
When you're trying to sort out syntax and bugs, leaving off the semicolon is very handy, but for this course, I want you to use `disp()` or `fprintf()` to output in anything you turn in.

The next question requires you to write and run a simple script. First master the following:
   i.   Open a new script in the Editor as described above.
   ii.   Copy the three lines of RNG initialization code above into the new script.
   iii. Add code that generates a row vector of 5 random numbers drawn from the uniform distribution (look up `rand()`, and see the examples in `matlab_demo_1.m`), and assign the vector to variable `v`.
   iv.   Save the script under a name of your choice, making sure it's in MATLAB's current working directory.

v.   Run the script by typing its name on the command line. You should see `v` appear in the Workspace window.

vi.   Now modify the script so that it:

         a.   Displays `v` in the Command Window.

         b.   Draws another 5 random numbers, assigns them to `v`, and displays the new `v` (use `disp()`).

Question 3) [10 points] Using elements from the script developed above, create a script to:

i.   Initialize a RNG with a random seed of 5.

ii.   Generate and display two random row vectors of length 6.

iii.   Generate and display a random 3 x 3 matrix.

iv.   Reinitialize the RNG with a random seed of 11.

v.   Repeat steps ii. and iii. [The results should be different from the first time.]

Run the script. Cut-and-paste the following into your answer document: i) the contents of your script, and ii) all the output displayed by your script in the Command Window.