E296MA
Dr. Gabriel Gomez
GSI Spencer Frank
Fall 2017

MATLAB Homework

Please turn in your code, and a single document with all problems. Videos should be turned in separately. Refer to MATLABTutorial1 document for details.

Problem 1

Perform a curve fit on the entire dataset (years 1901-2012) for the 1st, 2nd, and 3rd degree polynomial. For each of the three functions, extrapolate predictions from 2013 to 2050. Plot the three predictions and comment on which function you believe gives the most realistic predictions for the period 2012-2050. Turn in with Homework set 3.

Problem 2

Using the weight data (instead of height) create the four histograms, fit a gaussian distribution, and plot the sample mean vs. sample size (law of large numbers). Your figures should be similar to the height histogram and mean height vs. sample size figures above.

Problem 3

Take a colony of N_ants = 40. Leave all ant properties set as their default. You'll notice in running the movement simulation that the ants are not even distributed between the four food locations. However, this is not optimial, as only 10 ants can eat at one time at each location. This property was previously set antColony.maxAntsEating]=deal(10).

Your task is to add functionality to antDef with the objective of distributing the 40 ants evenly between each food source, so that 10 ants will be eating at each location. The figures above show what to expect when you do this right.

To do this you only need to focus on one part of the code. You will not need to edit any other part of the code if you do this right.

```
elseif magnitudeNearestFood<=antCurrent.vision %if ant is close to food, coun
    %COMMENT OUT THE THREE LINES BELOW FOR HOMEWORK
```

```
antCurrent.foundFood=foodsourcenumber;
antCurrent.friendDesire=0;
nNearestFood=vectorNearestFood/(magnitudeNearestFood);

% ADD NEW CODE HERE
```

When writing the new code, think about the following three tasks

Task 1: Quantify the number of ants eating at a particular location. This can be done by defining an 'eating' ant as one that is within the vision radius (i.e. less than 0.2 [m] from the food location). The 'eating' Ants should be assigned a label that corresponds to the food location (1,2,3, or 4).

Task 2: If the number of ants 'eating' is less than maxAntsEating, then do something like continue toward the food, and perhaps change foodDesire to 1, and friendDesire to 0. No one cares about friends, once they smell food right?

Task 3: If the number of ants 'eating' is equal to maxAntsEating, then do something else, like move away from the food and head in another direction.

These three tasks can be achieved with the following code structure

```
numAntsEating=???;
if numAntsEating < antCurrent.maxAntsEating
    nNearestFood=???
    antCurrent.foundFood=???;
    antCurrent.foodDesire=???;
    antCurrent.friendDesire=???;
elseif numAntsEating >= antCurrent.maxAntsEating
    nNearestFood=???
    antCurrent.foodloc(foodsourcenumber,:)=???;
end
```

The last line redefines the antCurrent.foodloc so that the ant no longer goes toward the food that has many ants eating it.

Turn in your code, a video of your ant colony, and the final tabulated distribution.