CogSci 109, Homework 6:
MIKE, Optimization, Gradient Descent, Error, and Uncertainty
(100 pts total plus 10 bonus possible)

Alex Simpkins

December 10, 2007

1 Description

Now that you can load data into matlab, visualize the basic information, and create simple mathematical models, you need to be able to evaluate those models. In lecture weve discussed the concept of error analysis and uncertainty. Weve also discussed optimization within the context of minimizing cost. Those two concepts will be the focus of this assignment.

You will perform linear function minimization (conjugate gradient descent) on data to determine the best behav- ioral model for a new life form. You will then answer a few questions regarding your results.

Please read this entire document before proceeding. This is a long document because the programming steps have been extensively documented and most of them given to you directly. So if you follow along with matlab and this pdf open simultaneously, you should hopefully find the programming part fairly straightforward. The questions are designed to help you think about the material and come up with meaningful answers.

Formatting Requirements: (10 points)

- Cover page with your name, the date, class, quarter, your section, and the homework number/title (2 pts.)
- Pages must be numbered (1pt.)
- No plots should be JPEGs. When using the *save-as* command in Matlab, use some vector-based graphics format (or one of more appropriate compression strategies)

such as PDF, EPS, etc when exporting figures from matlab to your document file (if you have any issues with this, email us or otherwise let us know). The plots must be clear and not blurry. (2 pts.)

- You may recycle printer paper which has been printed on only one side IF what is on the back does not bleed through to the front or in any way interfere with your assignment. Also the material on the reverse side should be clearly unrelated to this assignment. (1 pt.)
- You may NOT use lined paper (such as from a binder or college ruled paper) (1 pt.)
- All figures must have axis labels and a figure title. If there is more than one data line or there is a scatterplot of *'s and lines, you must use a legend to clarify which line corresponds to what information source. (1 pt.)
- If you have plots with more than one line, and since you are printing in black and white, you should use more than one linetype to set the lines apart. (1 pt.)
- Turn in a well commented listing of your matlab code in an Appendix at the end of your assignment paper, or alternatively, with each problem (1 pt.)

2 Background

Simple deterministic relationships can provide sufficient models to predict many behaviors of creatures with primitive or complex brains. Here we will explore the power of simple models to predict behavior. In this hypothetical case pretend you and your team have discovered a new species of life. It has not been classified yet (it is difficult to determine if it is an insect or mammal, possessing characteristics of both, thus it defies current classifications). The creature has been named M. I. K. E. (this stands for **Mechano-Intelligent Kinetic Exopod**). MIKE is easy to experiment with, since he (or she) loves food (especially pizza), is completely complacent and friendly with experimenters, and has EEG plugs evolved into his exoskeleton. And, as opposed to rats and monkeys often used in cognitive experiments, MIKE smells pleasant.

3 Problem

It appears MIKE is reacting to environmental stimuli (or we assume this is likely given our prior knowledge of MIKE from experiments). The question is, what is the relationship of MIKE's behavior to standard basic stimuli (which many types of life forms respond to) such as light and food?

You have devised and executed a simple experiment with the following 3D terrain map which was created for MIKE in the new rapid prototyping machine lab in the cognitive science department in order to help answer this question.

The experiment consists of two parts (you will only be dealing with experiment part I in this assignment). The first part consists of placing food on a location above the terrain at a particular location, under water, and placing MIKE in the water at a different location. MIKE excitedly advances to the food (he is equally at home in water or air). The second part of the experiment consists of you placing MIKE in a random location on the terrain, and turning on an LED light (which is at an angle he can view) at another location on the terrain.

In both cases you record MIKEs movements with motion capture equipment (CALIT2 just purchased a $400k motion capture system that records movements with over 20 cameras which identify and track infrared dots in 3D space, then output position coordinates to you as a matrix of numbers at a sample rate of 5kHz). You only track MIKE's 3D position on the terrain map from a marker located roughly on top of his body.

4 Modeling the movement

4.1 Description

A simple pilot study was previously performed, and it has been shown that MIKE has a simple feedback-driven behavior. Essentially what this means is that, in the case of attraction toward a stimulus, such as food, a good model of MIKE's behavior can be made by some equation mapping the error (distance between MIKE and stimulus) into action.

In the case of food, it appears that MIKE's movement is directly related to this error. After performing many experiments, you want to create a simple linear model predicting his estimated velocity relative to distance from food.

In the case of a light stimulus, it appears that MIKE has a movement which depends in a less trivial, but still predictable way on the distance from the light source. In this homework we will not explore this relationship.

4.2 Load the data (20 points)

4.2.1 Step1: Download the data

Download the data (the file is called hw6data.zip) from the assignments page. Unzip the file, and you will have one file. It should be called hw6_EXPER1_data.mat. Verify that this is true before proceeding.

4.2.2 Step2: Load the data into Matlab (20 points)

Load the data file called hw6_EXPER1_data.mat into Matlab using one of the methods discussed thus far in the class. Write down the steps you used to load the data (i.e., if you double clicked it, explain how you did it. If you used the import wizard or a matlab command, explain those methods). Include any code you may have used for this step here.

Type at the command window:

`whos`

You should see the following result in the matlab command window:

Name	Size	Bytes	Class	Attributes
e	1000x3	24000	double	
food	1000x3	24000	double	
start	1000x3	24000	double	
velocity	1000x1	8000	double	

e has 1000 observations of the difference between the food and start point for MIKE, organized into rows, and each column represents (x, y, z). The food and start matrices are also included, and were used to compute e by $e = food - start$. The velocity vector represents the velocity at each observation (row). The rows of velocity correspond to the rows of e, food, and start.

This is the experiment 1 data.

4.3 Plot the raw data (20 points)

Create a 3D plot of the data points for e using the `plot3()` command in matlab. Type `help plot3` in matlab to gain more information.

You should see a plot that has points scattered in 3D space (BE SURE TO PLOT USING THE '*' LINE FORMATTER - i.e. `plot3(x,y,z,'*')`). Include this plot in your paper as a printout, as well as the code to perform the plot.

4.4 Fit a simple first order model to the data (20 points)

Here we will use conjugate gradient descent to fit a simple model to the data, based on the equation for a straight line, as we've seen many times before. The equation will have the following form:

$$v = p_1 + p_2 * ||e||^2 \tag{1}$$

where the p's are the unknown parameters of the equation, $e = food_{location} - start_{location}$, and v is the expected (magnitude of the) velocity of MIKE. In this experiment MIKE was swimming and so could move in 3-dimensions, but we will first consider his square distance to the food using the squared norm ($|| \cdot ||^2$).

Let's write the code and perform the fit.

We'll go through the code and write it here. Matlab code will be written in `typwriter red font like this.`

NOTE: Because of the way LaTeX typesets fonts, if you copy and paste the code below you will probably have problems - I am including the code here for you to use. It does NOT have comments at a certain point, and it will be your job (in the coming questions) to comment the code and explain it. Don't worry, I'm here to help you through it:)

Let's start by closing all the extra open documents:

```
close all
```

Then we have the data from EXPERIMENT 1 in memory already, so next we begin to setup our problem. Just like with the least squares, we're going to try to solve

$$Ak = v \tag{2}$$

where v is the velocity variable, A is a matrix of ones and $||error||^2$ inputs. Well do a linear fit, just like least squares, but using gradient descent. But before we can, for a gradient descent algorithm to work on an over-determined system (ie more data points than unknown parameters) we need to transform the system into an equivalent one which is not over-constrained (a perfectly constrained system). We do this by multiplying both sides of the equation on the left by A^T. This will also make our A matrix positive definite. In equation form this will give us

$$A^T A k = A^T v \tag{3}$$

and in code...

```
en= (sum((e.^ 2)'))';
```

```
A=[ ones(length(e(:,1)),1) en];
```

```
M = A'*A;
```

```
v = A'*velocity;
```

Now we clear the memory of any previous parameter saves:

```
clear res_save k_save k;
```

Then setup our stopping criteria - a tolerance...

```
epsilon = 1e-11;
```

and maximum number of iterations.

```
itmax = 1000;
```

Then we set aside enough memory for the parameters

```
k=zeros(size(v));
```

And finally begin our iterations. We will begin with an initial guess, and modify our guess iteratively to converge to the solution, or at least to a close approximation:

```
for iter=1:itmax
    r=v-M*k;
    res=r'*r;
    res_save(iter)=res;
    k_save(:,iter)=k;
    if(res < epsilon || iter==itmax),break; end
    if (iter==1)
        p=r;
    else
        beta=res/res_save(iter-1);
        p=r+beta*p;
    end
```

```
    alpha=res/(p'*M*p);
    k=k+alpha*p;
end
disp('success - goal achieved!')
```

4.4.1 Comment the code (10 points)

Comment each line of the code, explaining, based on the numerical methods chapter on minimization and specifically the section on conjugate gradient descent, what the line does. A simple one line explanation is all that is needed (like in the book). Do not copy exactly what the book says, or what I said in class. It should be at least slightly expressed in your own words.

Include your code with the problem number.

4.4.2 Plot the parameter evolution over the objective function (10 points)

Create a plot of the objective function (I suggest using the contour command, which is somewhat similar to the pcolor command - type `help contour` in matlab for more info). The objective function we're minimizing is

$$J = \frac{1}{2}k^T M k - v^T k \tag{4}$$

So we'll plot that over the same range as our parameters. Once again, as with the least squares fits in previous homeworks, we have two parameters. This lends itself to a pcolor or contour plot of the objective function. The code to compute J is given below (all you have to do is plot J in a contour or pcolor plot). You can change the contour plot range if appropriate by changing ZLIM

```
en2=(sum((e.^ 2)'))';
M = [ones(length(e(:,1)),1) en2]'*[ones(length(e(:,1)),1) en2];
v2 =[ones(length(e(:,1)),1) en2]'*velocity;

ZLIM=6;
z1=(-ZLIM:(2*ZLIM)/(100-1):ZLIM)';
z2=(-ZLIM:(2*ZLIM)/(100-1):ZLIM)';
[Z1,Z2]=meshgrid(z1,z2);
```

```
n=length(z1(:,1));

for j=1:n,
    for i=1:n,
          z=[Z1(i,j) ;Z2(i,j)];
          J(i,j) = (1/2)*z'*M*z - v2'*z;
    end
end
```

Plot the evolution of the parameters in the same plot over the objective function (use `hold on` and plot with the `plot()` command, not the contour command for the variables, only use `contour` for the J, you will also need to use the meshgrid command to scale the locations of the two axes of the J), starting with the initial guess (recall the zig-zag line we saw in lecture plotted over the objective function - here the 'movement' pattern will be straighter). The plot should include the usual - labels, titles, color scales.

4.4.3 Characterize the 'goodness of fit' (10 points)

Compute the norm-based prediction error by calculating a predicted velocity:
`velocity_pred=k(1)+k(2)*en2`

Then the norm of the error between predicted and actual:
`Err_n = norm ( velocity - velocity_pred )`

Now look at the correlation between the squared distance and the velocity magnitude.
`C = corr( en2, velocity)`

and list the results in a table you create in word or a similar program.

4.5 Questions (20 points)

4.5.1 Question 1 (10 points)

Does it appear that there is a linear relationship between the squared distance between food/start point and velocity of MIKE? Support your answer with the numbers computed previously.

4.5.2 Question 2 (5 points)

What is an advantage of conjugate gradient descent over least squares in the case of a more general equation (with possibly nonlinear in the parameter terms)?

4.5.3 Question 3 (5 points)

What is another technique we learned for function minimization that you could use in the general case, in addition to the various forms of gradient descent, and roughly how could you apply it here?

4.6 BONUS: Least squares (10 points)

Compute a least squares linear fit, and compare the parameters (in a table made in word or a similar program). Why are they so similar (briefly)?

—————END OF ASSIGNMENT—————