CogSci 109: Lecture 12

Wednesday Oct 31, 2007

Nonlinear least squares, definition of regression (linear vs. nonlinear regression), correlation and interpretation

Outline for today

- Announcements
  - **Homework 4 assigned late wed night, due Wed of next week**
  - **Midterm next Friday?**
- Clarifications about linear equations vs. strict linearity
- Linear regression
  - **Introduction to nonlinear least squares**
- Correlation coefficient

Clarifications about linear equations vs. linear systems

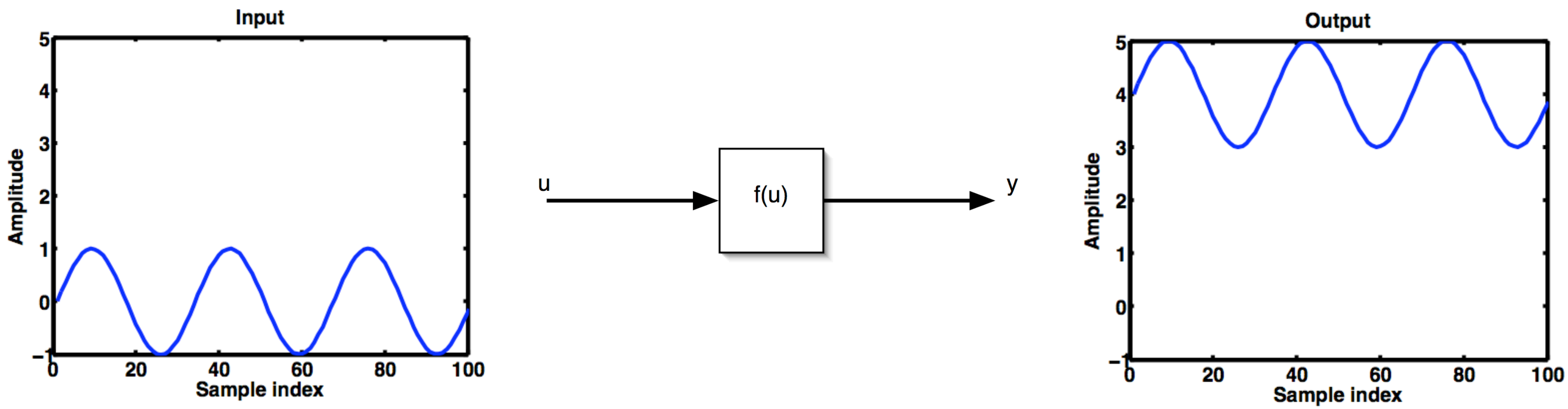
- Last time we referred to y=mx+b as a **linear equation**
  - **A linear equation is commonly referred to as an equation whose curve is represented by a constant coefficient or a constant times a variable whose power is 1**
  - **Commonly represented as a 1st degree polynomial (with the highest order variable (the power of x) is 1)**
  - **But…**

Clarification about linear equations vs. linear systems

- This is not the same as a **linear function** or **linear map**
- You may note that **y=mx+b**, if **b** is nonzero, does not satisfy both tests for linearity ($f(mx_1+nx_2)!=mf(x_1)+nf(x_2)$)
  - **It is additive, but not homogeneous**
  - **It is *commonly* referred to as a linear equation because it defines a straight line in Cartesian coordinates**
  - **Must be differentiated from strict linearity**
    - This is referred to as an *affine transformation* (or *affine map*) when b is nonzero
    - **Affine transformations** are more general than linear transformations - a linear transformation followed by a translation (b term)

Nonlinear (or is it?) example:

- Is this linear?



- The function instantaneously jumps from being centered about zero to being centered about 4! How can that be linear? It’s NOT!!!

Why present this detail?

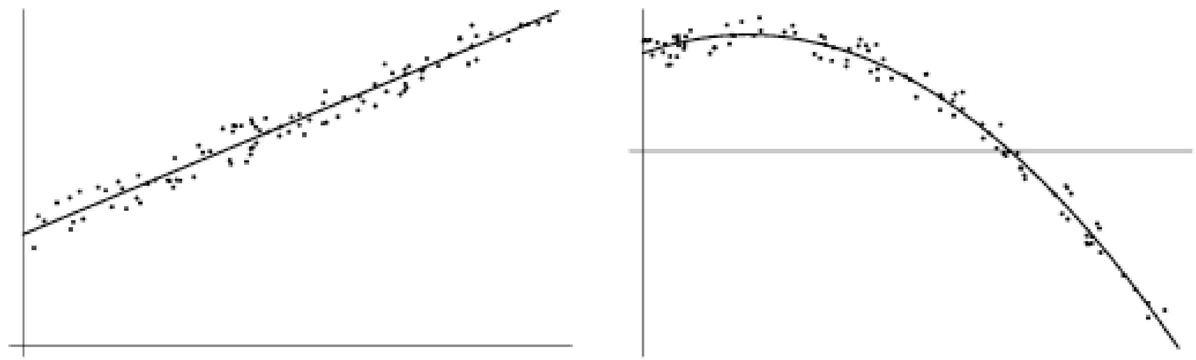
- To drive home a few points
  - **Linearity is a tricky thing to intuitively grasp**
  - **Linearity is subtle,and may not have been clearly defined in earlier math classes**
  - **Many systems that appear nonlinear can be represented by a linear system, but ALSO, many systems that *appear* linear may also be in fact *nonlinear* when considered from the right perspective**
  - **Just remember that test: additivity and homogeneity**

$$f(mx_1 + nx_2) =? mf(x_1) + nf(x_2)$$

  - **The study of spaces and transformations is useful and broadening to the mind - sometimes thinking in a different type of space allows one to solve a problem**

Nonlinear least squares

- What if the data isn't linear?!?
- Still can be done with linear regression!



- We can fit a higher order polynomial which is nonlinear in the independent variables but linear in the unknown parameters, as shown above, right

Nonlinear least squares (II)

- How do we do this?
  - **First consider an nth degree polynomial**

$$y = a_0 + a_1 x + a_2 x^2 + \ldots a_n x^n$$

    - Nonlinear equation
    - Linear in the parameters
  - **We want to determine the *a*'s that make the curve most closely pass through a set of data**
  - **We do this in the same way as before in matlab**

Nonlinear least squares (III)

- Consider that we have some data, as before:

```
x=[1 3 2]
y=[2 4 3.5]
plot(x,y,'*')
```

- And we want to fit an equation with a nonlinear term:

$$y = mx + nx^2 + b$$

Nonlinear least squares (IV)

- How do we solve this? Well if we pre-compute x^2 for our data, we have the following problem:

```
2   =m(1) + n(1) + b
4   =m(3) + n(9) + b
3.5 =m(2) + n(4) + b
```

- Which we can write in matrix form:

```
[2    =  [ 1 1 1    [m
 4         3 9 1     n
 3.5]      2 4 1 ]   b]
```

```
A=[1 1 1; 3 9 1; 2 4 1]
y= [ 2; 4; 3.5]
```

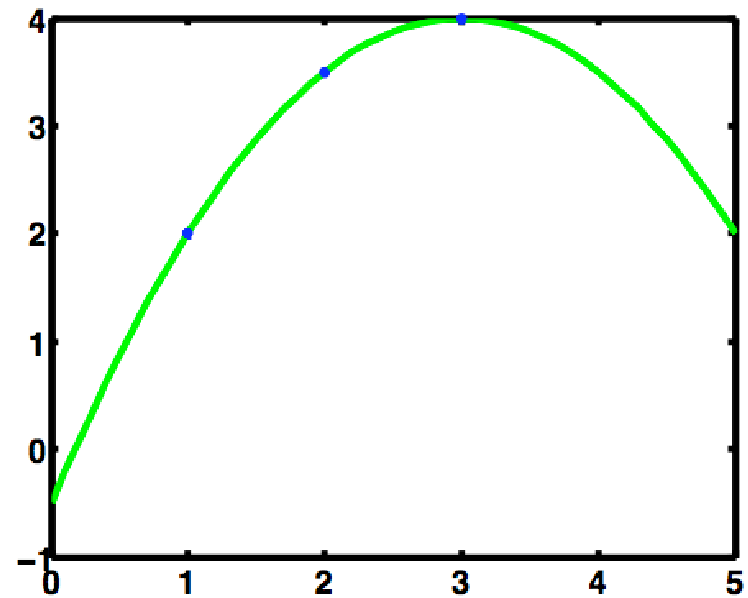
- Again we can solve this with:

```
mnb= A\y
```

Let’s plot our fit against our data

- In order to do that we just create new x-data (so we have information not only at the points we used for our fit), and plug it into the equation we just found the parameters for

```
xp=0:.1:5;
yp=mnb(1)*xp+mnb(2)*xp.^2 + mnb(3);
plot(xp,yp,'g')
```

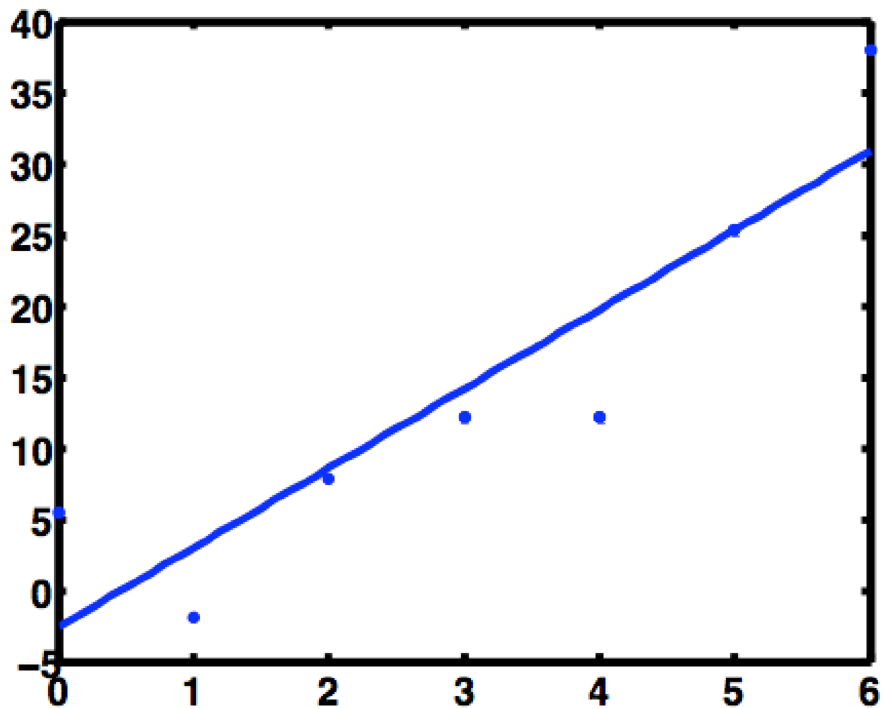


Taking it further...

- Let's do this for a larger dataset:

```
x=0:6
y=x.^2 + 3*randn(1,length(x))
plot(x,y,'*')
```

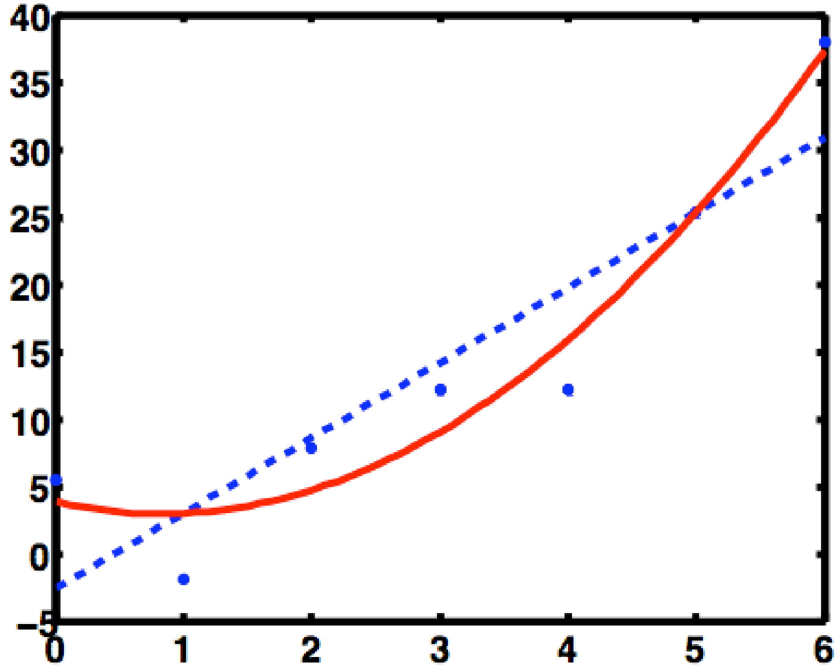
```
A = [ x' ones(length(x),1)]

mb= A\y'

xp=0:.1:6
yp=mb(1)*xp +mb(2)

hold on
plot(xp,yp)
```



Taking it further (II)...

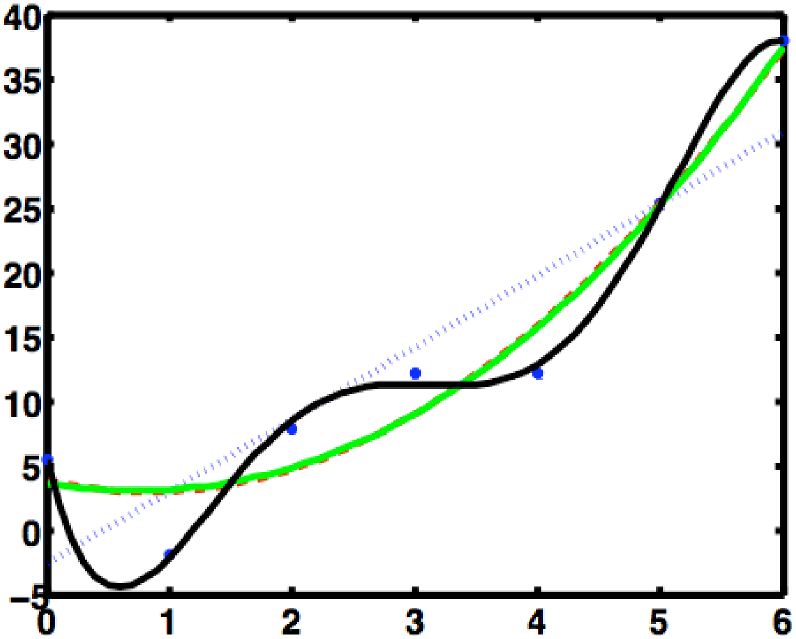
- Let's try fitting a nonlinear polynomial:

```
A = [ x' x.^2' ones(length(x),1)]
mnb=A\y'

xp=0:.1:6
yp=mnb(1)*xp +mnb(2)*xp.^2+ mnb(3)

plot(xp,yp,'r')
```



Going yet further with the fit...

- Let's try to fit



y=mx+nx^2 +px^3 + qx^4 + rx^5+b

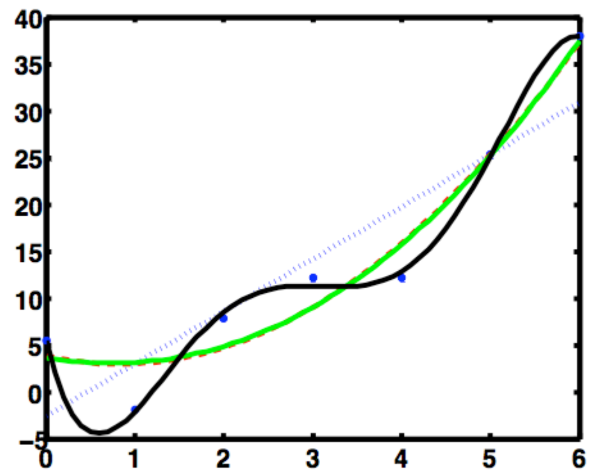
```
A = [ x' x.^2' x.^3' x.^4' x.^5' ones(length(x),1)]
mnpqrb=A\y'
```

```
xp=0:.1:6
yp=mnpqrb(1)*xp +mnpqrb(2)*xp.^2+ mnpqrb(3)*xp.^3 +…
        mnpqrb(4)*xp.^4 + mnpqrb(5)*xp.^5 + mnpqrb(6)
```

```
plot(xp,yp,'k')
```

Overfitting...

- We refer to a fit such as the black line as **overfitting**
  - **You are no longer fitting the system's interrelationships, you are fitting noise**
  - **More on this later**
    - But one easy technique for reducing overfitting is to remove certain points which appear to be deviating in a non-systematic way, and test the fit again
    - A nice application of machine learning



- Sometimes a model which is too complex provides a worse model of what is ocurring than a simpler model for this reason