Curve fitting

Fitting parabolic and exponential curves

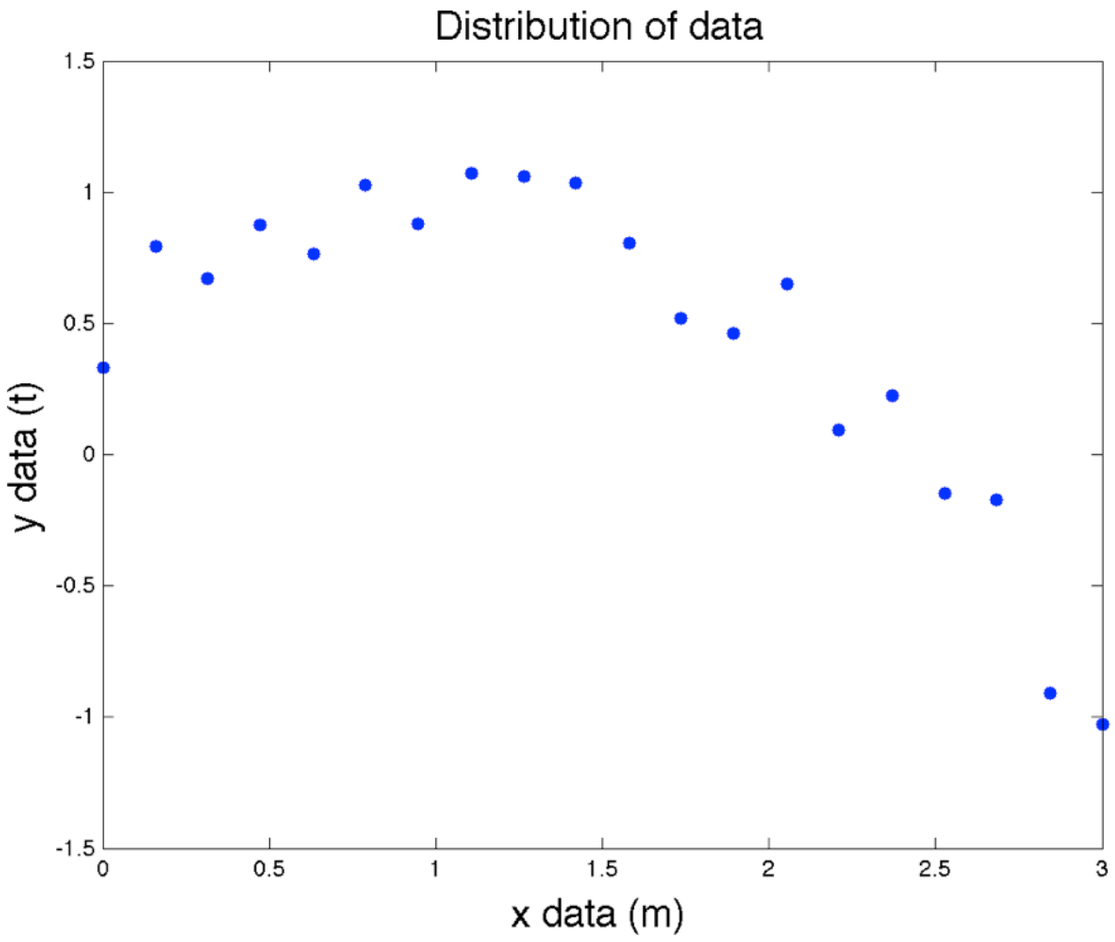
Wednesday, March 13th

Math 365

Week #8

Linear least squares

What if our data looks like this?



A linear equation may no longer be a good model of the underlying physical process that generated the data.

Parabolic model

A better fit might be

$$y = ax^2 + bx + c$$

Question :

How do you know what is a good model?

Answer :

Depends on what you are trying to do. In many cases, you have an understanding of the physical processes that produced the data, and so you can develop a model based on the physical assumptions.

Fitting a parabola

Again, we are seeking model parameters. In this model, the unknown coefficients are a, b, and c.

$$y = ax^2 + bx + c$$

Just as in the linear case, we write down an expression for each data point, assuming that the data point "solves" the model :

Given data points (x_i, y_i), we assume

$$y_i = ax_i^2 + bx_i + c, \qquad i = 1, 2, \ldots, N$$

Notice that the system is still *linear* in a,b, and c.

Fitting a parabola

We can express this as a linear system in a, b, and c :

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}$$

Again, the system does not have a solution in general. We can however, find a best fit solution, again using the normal equations.

$$\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}$$

Loading and plotting the data

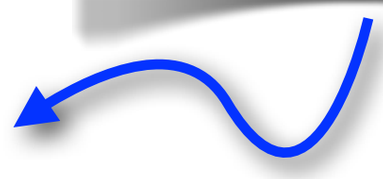
Suppose we have a data file whose columns contain the x and y data :

```
% xy_data.dat
    % x                 % y
0.0000000e+00    3.3109115e-01
1.5789474e-01    7.9257164e-01
3.1578947e-01    6.7153550e-01
4.7368421e-01    8.7576241e-01
6.3157895e-01    7.6495733e-01
..........
```

Load and plot the data points :

```
% Load and plot data
xy_data = load('xy_data.dat');
x = xy_data(:,1);
y = xy_data(:,2);
plot(x,y,'.','markersize',20);
```

Increase the size of the symbols so they show up on the plot



Solving the normal equations

We can construct the linear system as

```
% Set up linear system
N = length(x);
A = [x.^2 x ones(N,1)];
b = y;
```

One could solve the normal equations directly :

Expensive!

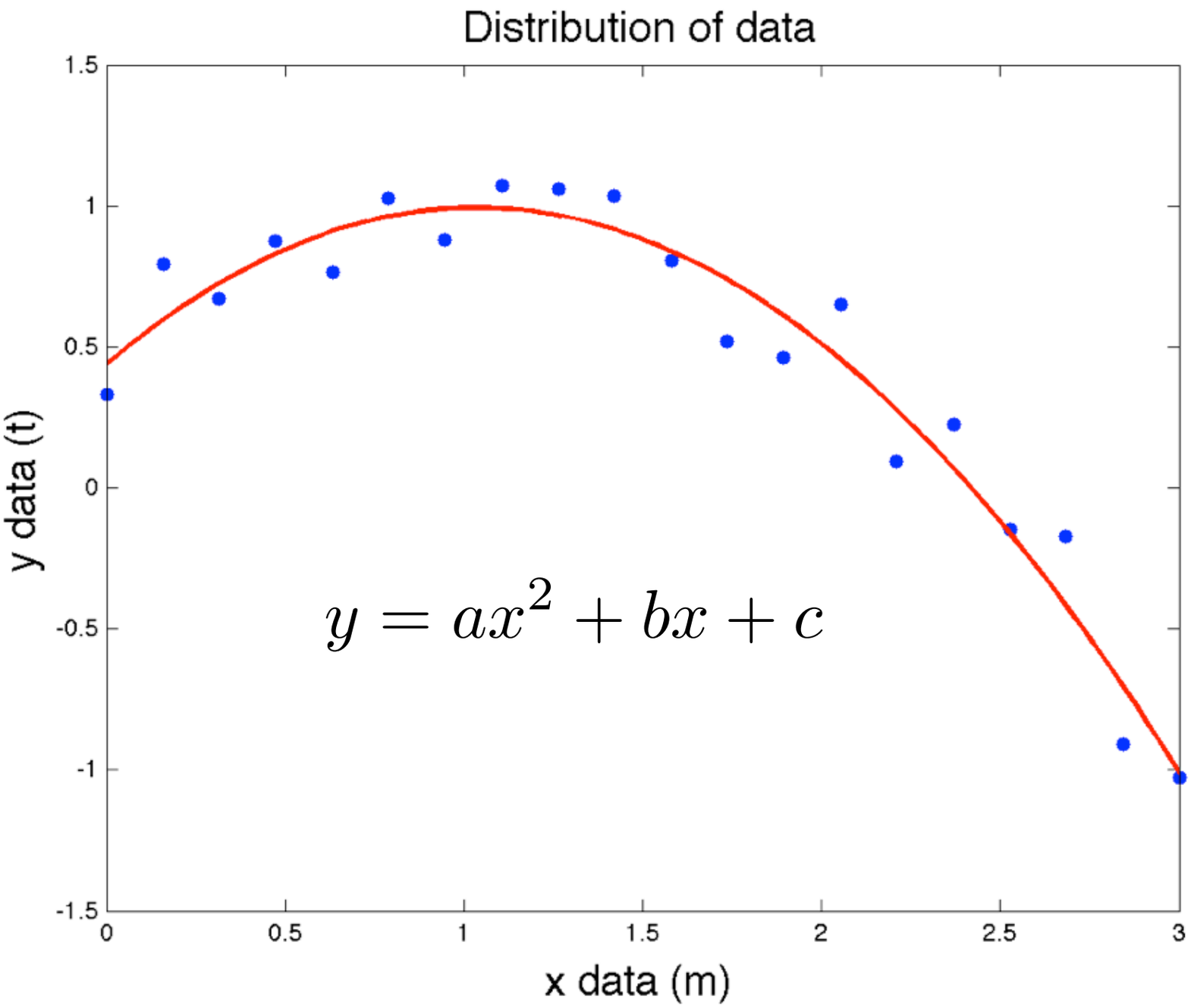
~~`coeffs = inv(A'*A)*A'*b;`~~

But it is much better (and easier to remember!) to do this:

Matlab uses a QR factorization

```
coeffs = A\b;
```

Best fit parabola




```
a =    -0.519047; b =     1.074011; c =     0.438339
```

Parabola fit - complete code

```
xy_data = load('xy_data.dat');
x = xy_data(:,1);
y = xy_data(:,2);
plot(x,y,'.','markersize',20);

% Set up linear system
N = length(x);
A = [x.^2 x ones(N,1)];
b = y;
coeffs = A\b

% Plot parabola
a = coeffs(1);
b = coeffs(2);
c = coeffs(3);
xs = linspace(0,3,100)';
ys = a*xs.^2 + b*xs + c;
hold on;
plot(xs,ys,'r','linewidth',2);
xlabel('x data (m)','fontsize',18);
ylabel('y data (t)','fontsize',18);
title('Distribution of data','fontsize',18);
```

Fitting general polynomials

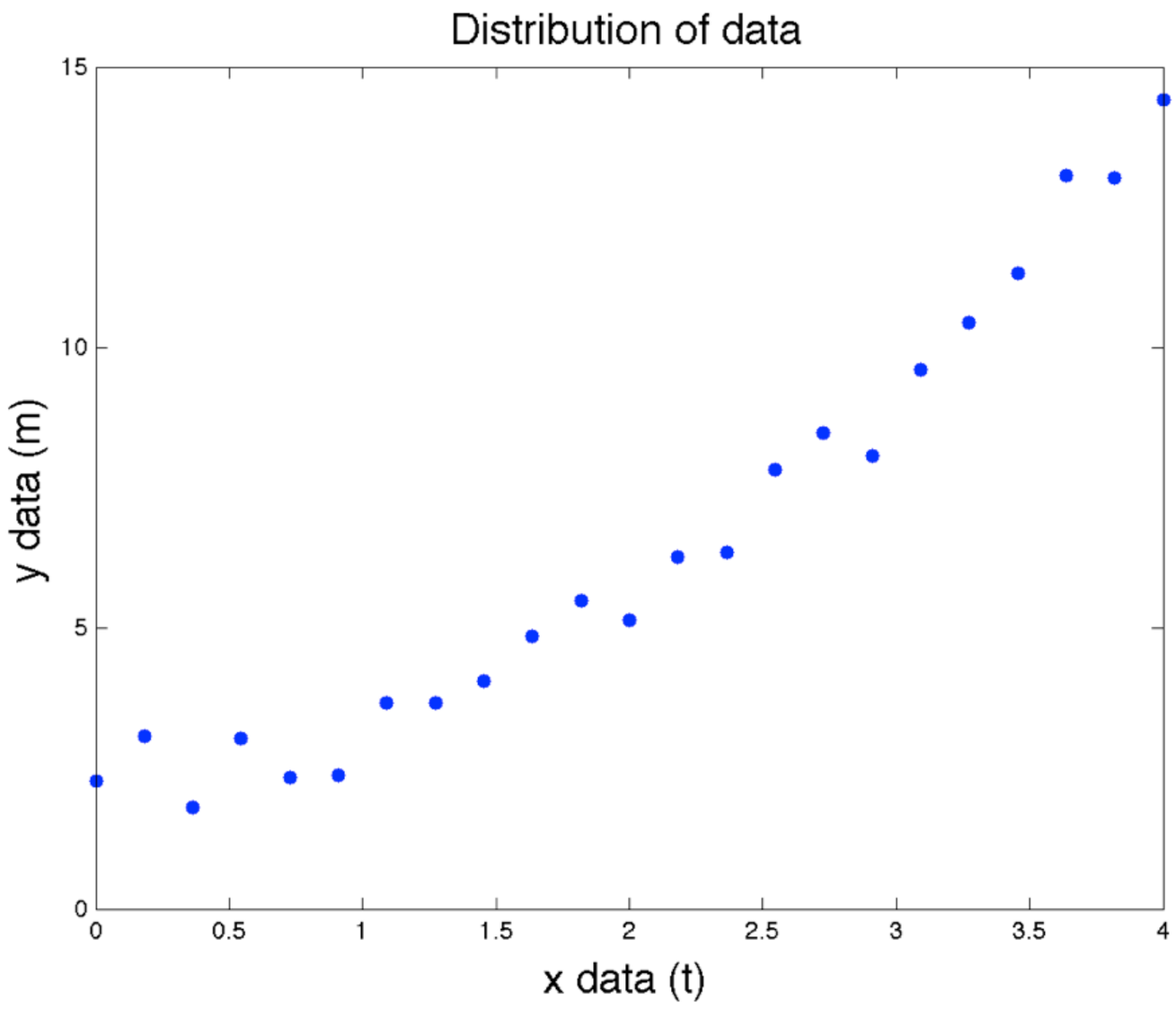
General polynomials can be fit using linear least squares.

$$y = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

where it is assumed that the number of data points N is at least $n + 1$.

One must be careful, however, as the linear systems will become increasingly ill-conditioned as the n increases.

Fitting an exponential curve



Many distributions of data may be better fit by an exponential curve. Can we still use linear least squares?

Fitting an exponential curve

An exponential model is :

$$y = ae^{bx}$$

But the model is not linear in the parameters a and b. How are we going to use linear least squares?

The trick is to take the natural logarithm of both sides to get :

$$\ln(y) = \ln(a) + bx$$

The model is now linear in b and $\ln(a)$.

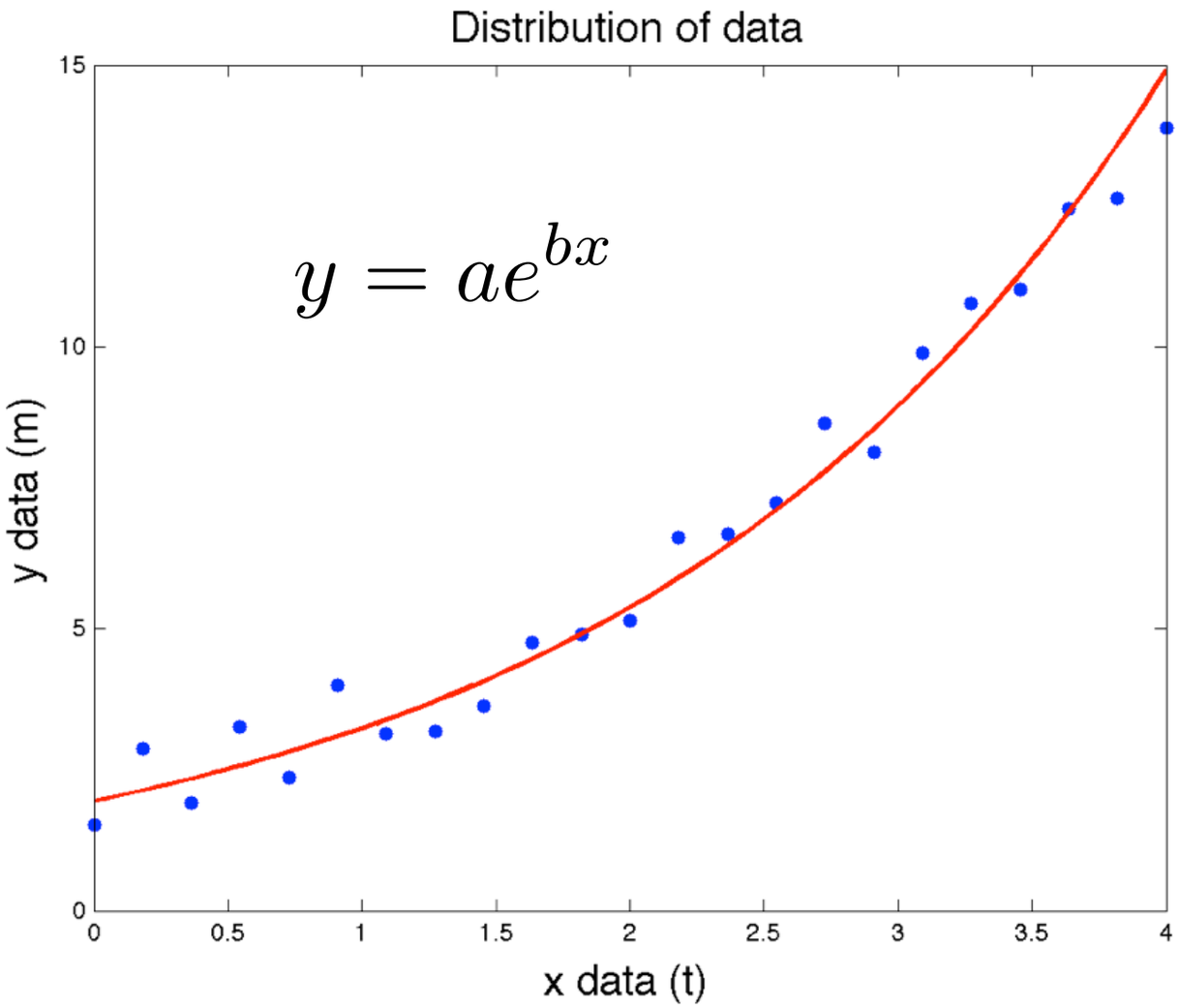
Fitting an exponential curve

Again, we can set up a *linear* system :

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} b \\ \widetilde{a} \end{bmatrix} = \begin{bmatrix} \ln(y_1) \\ \ln(y_2) \\ \ln(y_3) \\ \vdots \\ \ln(y_N) \end{bmatrix}, \qquad \widetilde{a} = \ln(a)$$

We can then compute $a = e^{\widetilde{a}}$.

Fitting an exponential curve




```
Coefficients : a =     1.944548; b =     0.509077
```

Fitting an exponential curve

```
% First, "generate" our data to fit :
N = 23;   % Number of data points
x = linspace(0,4,N)';
y = 2*exp(0.5*x) + (-1 + 2*rand(N,1));  % add a random
                                        % number in [-1,1]

% Plot our data points
plot(x,y,'.','markersize',20);   % Make big dots
hold on;

% Now find the best fit exponential curve
A = [x ones(N,1)];
rhs = log(y);    % rhs = right hand side
coeffs = A\rhs;

b = coeffs(1);
a = exp(coeffs(2));

xs = linspace(0,4,100);
ys = a*exp(b*xs);
axis([0 4 0 15]);
plot(xs,ys,'r','linewidth',2);
xlabel('x data (t)','fontsize',18);
ylabel('y data (m)','fontsize',18);
title('Distribution of data','fontsize',18);
```

Download the code

lecture_10.m