Lesson 17: Least Squares Fit

1 Introduction

There are occasions when you are given a set of data, either from an experiment or from a complicated numerical calculation, listing the corresponding values for two variables. For example, the data might list the pressure P of a gas at various temperatures T, or the electric potential V at various points on the x–axis, or the intensity I of a laser at various times t. We would like to know the functional relationship between these variables. In some cases we may have a theoretical reason to expect a certain relationship between the variables; in other cases the data itself might suggest a type of functional relationship. However, the data will always contain errors, causing the data points to be scattered. We need a rational way determine the functional relationship between these variables; that is, we need a way to choose a function that best fits the scattered data.

2 Example

As a concrete example, consider an experiment in which we drop a ball and record its speed as function of time. From this data we want to find the acceleration due to gravity. The dataset *freefall.data* can be found on the moodle course page. The first column is a list of times at which the measurements were made, the second column is the corresponding speeds.

Download *freefall.data* and save a copy in the same directory (folder) as your python program. The data plot should look like Figure 1.

> Exercise: Create a code that reads the freefall data and plots the speed versus time. Add the line $s = 9.8\,t$ to the the graph, showing the behavior we should expect.

In this example we know the relationship between speed and time, because we know the acceleration due to gravity is the constant $g = 9.8\,\mathrm{m/s^2}$. Let's pretend that we don't know the value of g, and we want to use the data to find it. We do know enough to expect a linear relationship between speed s and time t, and apart from experimental errors the data do seem to follow a linear trend. So our goal will be to find the straight line that comes closest to fitting the data.

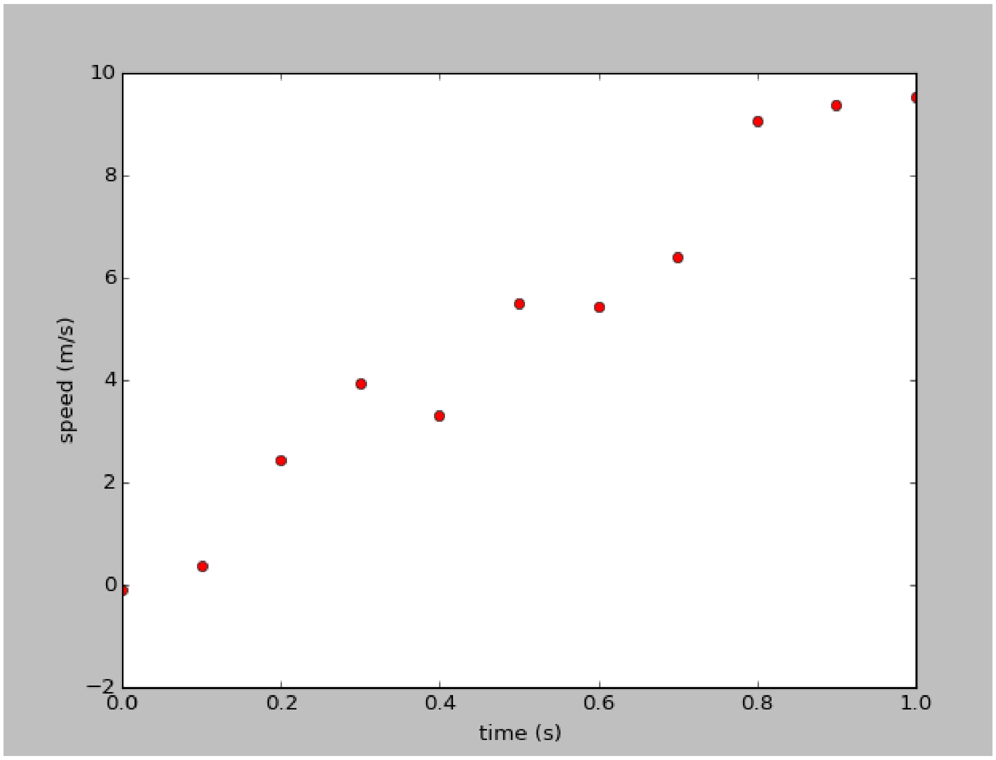

Figure 1: Experimental data showing speed versus time for a falling object.

3 Least squares

Let x_i, y_i denote the data, where $i = 1, \ldots, N$. We are looking for a function $f(x)$ that best fits this data. The definition of "best fit" is the function that minimizes the sum of squares of differences between the function values $f(x_i)$ and the data values y_i. That is, we define

$$S = \sum_{i=1}^{N} (f(x_i) - y_i)^2 \tag{1}$$

and look for the function $f(x)$ that minimizes S.

This is the "least squares" method, and S is called the "L_2–norm". Note that the differences are squared so that each error contributes a positive amount to S. Another reasonable approach, which is sometimes used, is based on the "L_1–norm". The L_1 norm is defined by $S = \sum_{i=1}^{N} |f(x_i) - y_i|$. This is the "least absolute deviation" method of defining a best fit function.

The first step is to guess the form of the function $f(x)$. Let's assume a linear relationship, so the function has the form $f(x) = ax + b$. The L_2 norm becomes

$$S = \sum_{i=1}^{N} (ax_i + b - y_i)^2 \tag{2}$$

Note that S is a quadratic function of the constants a and b. The minimum of S satisfies

$\partial S/\partial a = \partial S/\partial b = 0$, which implies

$$\sum (ax_i + b - y_i)x_i = 0 \tag{3a}$$

$$\sum (ax_i + b - y_i) = 0 \tag{3b}$$

(We'll drop the limits on the summation signs for notational simplicity.) Noting that $\sum 1 = N$, these equations simplify to

$$a \sum x_i^2 + b \sum x_i = \sum x_i y_i \tag{4a}$$

$$a \sum x_i + bN = \sum y_i \tag{4b}$$

This is a simple system of linear equations for the two unknowns, a and b. In matrix notation, we have

$$\begin{pmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum x_i y_i \\ \sum y_i \end{pmatrix} \tag{5}$$

The solution is easily found with numpy's `solve()` function,

```
import numpy as n
A = n.matrix([[a11,a12],[a21,a22]])
r = n.matrix([[r1],[r2]])
soln = n.linalg.solve(A,r)
a = soln[0,0]
b = soln[1,0]
```

with appropriate values used for the entries of `A` and `r`.

Homework: Write a program using the least squares method to determine the best fit line for the data from *freefall.data*. Plot the data and your best fit line on the same graph. What is the predicted value for the acceleration due to gravity?

4 Curve Fitting

Least squares analysis can be applied to any class of functions $f(x)$. For example, consider the data from the graph of Figure 2. The data appear to follow a sin–curve relationship, but we don't know the amplitude, frequency or phase. We can carry out a least–squares analysis to fit this data to a function of the form

$$f(x) = a \sin(bx + c) \tag{6}$$

That is, we solve for the parameter values a, b and c that minimize $S = \sum_{i=1}^{N} (a \sin(bx_i + c) - y_i)^2$.

This type of "curve fitting" is so useful that the module `scipy` includes the command `curve_fit` to carry out the analysis. You can import the command with

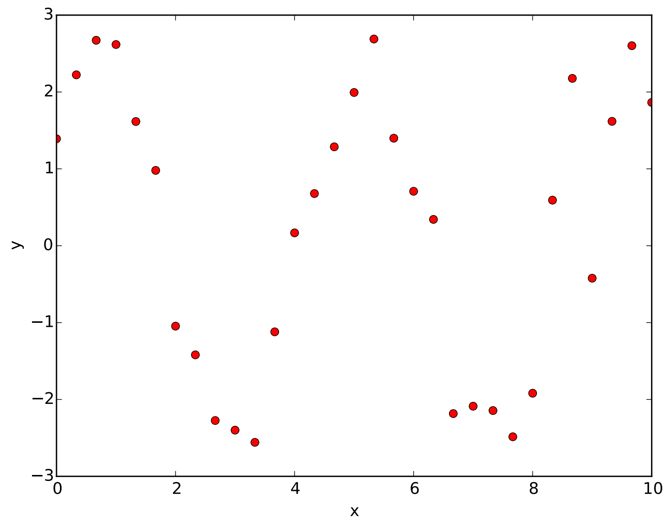

Figure 2:

```
from scipy.optimize import curve_fit
```

To use this command, you will need to create a definition

```
def func(x,a,b,c):
```

for the function (6). The first argument is the independent variable, and the remaining arguments are the parameters (in this case, a, b and c). Let's say the data are contained in arrays `xdata` and `ydata`. Then the command

```
par, con = curve_fit(func,xdata,ydata)
```

will return the least–squares values of a, b and c in the parameter array `par`. The array `con` contains further information that you can explore on your own.

> Homework: Write a code that will import the data file *sincurvedata* and carry out a least–squares fit to a function of the form (6). Plot a graph showing the data points, as well as the best fit curve. What values did you get for the amplitude, frequency and phase?