PHY371 ProjectIII
Least-squares fitting and the photoelectric effect *

Instructor: Ying Tang
Department of Physics
Gordon College

September 23, 2014

1 Introduction

It is a common situation in physics that an experiment produces data that lies roughly on a straight line, like the dots in this figure:

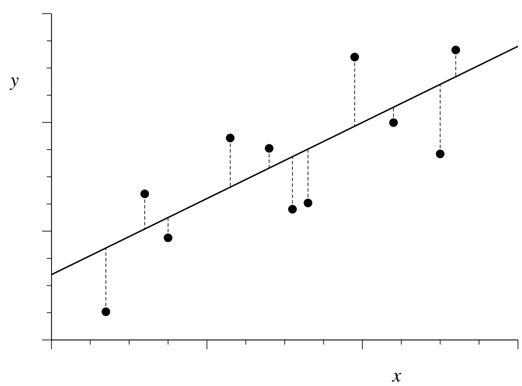

Figure 1: An example of the least-square fit.

The solid line here represents the underlying straight-line form, which we usually do not know, and the points representing the measured data lie roughly along the line but do not fall exactly on it, typically because of measurement error.

The straight line can be represented in the familiar form $y = mx + c$ and a frequent question is what the appropriate values of the slope m and intercept c are that correspond to the measured data. Since the data do not fall perfectly on a straight line, there is no perfect answer to such a question, but we can find

*This project is based on Exercise 3.8 in *Computational Physics* by Mark Newman the straight line that gives the best compromise fit to the data. The standard technique for doing this is the *method of least squares.*

Suppose we make some guess about the parameters m and c for the straight line. We then calculate the vertical distances between the data points and that line, as represented by the short vertical lines in the figure, then we calculate the sum of the squares of those distances, which we denote χ^2. If we have N data points with coordinates (x_i, y_i), then χ^2 is given by

$$\chi^2 = \sum_{i=1}^{N} (mx_i + c - y_i)^2 \tag{1}$$

The least-squares fit of the straight line to the data is the straight line that minimizes this total squared distance from data to line. We find the minimum by differentiating with respect to both m and c and setting the derivatives to zero, which gives

$$m \sum_{i=1}^{N} x_i^2 + c \sum_{i=1}^{N} x_i - \sum i = 1^N x_i y_i = 0, \tag{2}$$

$$m \sum_{i=1}^{N} x_i + cN - \sum_{i=1}^{N} y_i = 0. \tag{3}$$

For convenience, let us define the following quantities:

$$E_x = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad E_y = \frac{1}{N} \sum_{i=1}^{N} y_i, \quad E_{xx} = \frac{1}{N} \sum_{i}^{N} x_i^2, \quad E_{xy} = \frac{1}{N} \sum_{i=1}^{N} x_i y_i \tag{4}$$

in terms of which our equations can be written

$$mE_{xx} + cE_x = E_{xy}, \tag{5}$$

$$mE_x + c = E_y. \tag{6}$$

Solving these equations simultaneously for m and c now gives

$$m = \frac{E_{xy} - E_x E_y}{E_{xx} - E_x^2}, \quad c = \frac{E_{xx} E_y - E_x E_{xy}}{E_{xx} - E_x^2} \tag{7}$$

These are the equations for the least-squares fit of a straight line to N data points. They tell you the values of m and c for the line that best fits the given data.

2 Project

In this project, write a python program to do the following:

1. In the on-line resources you will find a file called "Project3_millikan.txt" (`https://github.com/IP4CS/Projects/tree/master/Project3`). The file contains two columns of numbers, giving the x and y coordinates of a set of data points. Write a program to read these data points and make a graph with one dot or circle for each point.

2. Now call the $curve\_fit$ module from *scipy.optimize* package to fit the data points and find the values of m and c ($let's$ call them m_1 and c_1). You should print out the m_1 and c_1 to the screen.

3. Write code that goes through each of the data points in turn and evaluates the quantity $mx_i + c$ using the values of m_1 and c_1. Store these values in a new array or list, and then graph this new array, as a solid line, on the same plot as the original data. Label this fit line *'Scipy Curve_fit'*.

4. Add code to your program, before the part that shows the graph [i.e. pylab.show()], to calculate the quantities E_x, E_y, E_{xx}, and E_{xy} defined above, and from them calculate and print out the slope m and intercept c ($let's$ call them m_2 and c_2) of the best-fit line.

5. Similar to step 3, write code that goes through each of the data points in turn and evaluates the quantity $mx_i + c$ using the values of m_2 and c_2. Store these values in a new array or list, and then graph this new array, as a solid line in another color, on the same plot as the original data and fitting curve found by python module. Label this new fit line *'Least Squares Fitting'*.

6. Compute χ^2 defined in Eq.1 for fittings done by python module and the least-square fitting respestively. Print out two χ^2s to screen with 16 digits after the decimal point.

3 Report

Use LaTeX to write a short (less than 3 pages) scientific report. Your report should include:

1. Introduction
   Describe the problem that you are working on.

2. Procedure
   Describe the methods that you use to solve this problem.

3. Results and Discussions

   - Show the graph with the original data and two fitting curves (plotted by python module and least-square fit method). Describe the graph in detail.

- Show χ^2 of two fitting curves. χ^2 is usually used to describe the goodness of fit. From what you have calculated, can you say something about the goodness of your fits? Which fitting method works better on this set of data?
- The data in the file "Project3_millikan.txt" are taken from a historic experiment by Robert Millikan that measured the photoelectric effect. When light of an appropriate wavelength is shone on the surface of a metal, the photons in the light can strike conduction electrons in the metal and, sometimes, eject them from the surface into the free space above. The energy of an ejected electron is equal to the energy of the photon that struck it minus a small amount ϕ called the work function of the surface, which represents the energy needed to remove an electron from the surface. The energy of a photon is $h\nu$, where h is Plancks constant and ν is the frequency of the light, and we can measure the energy of an ejected electron by measuring the voltage V that is just sufficient to stop the electron moving. Then the voltage, frequency, and work function are related by the equation.

  $$V = \frac{h}{e}\nu - \phi \tag{8}$$

  where e is the charge on the electron. This equaiton was first given by Albert Einstein in 1905.

  The data in the file "Project3_millikan.txt" represent frequencies ν in hertz (first column) and voltages V in volts (second column) from photoelectric measurements of this kind. Using the equation above and the program you wrote, and given that the charge on the electron is $1.602 \times 10^{19} C$, calculate from Millikans experimental data a value for Plancks constant. *Compare your value with the accepted value of the constant, which you can find in books or on-line.* You should get a result within a couple of percent of the accepted value.

4. Conclusion
   Summarize your problem, methods and results in a few words.

4 Submission

Please submit your python program (.py file) and your report (.tex and pdf files) to ying.tang at gordon.edu before **11:59pm on Sunday 10/5**. A late submission will cause a 50% deduction of your grade.