PHY 410
Homework Assignment 1

Han Wen

Person No. 50096432

September 2, 2014

Abstract

The goal of this assignment was to use numerical method to analyse some astronomical data to obtain Hubble's constant from various ways.

Contents

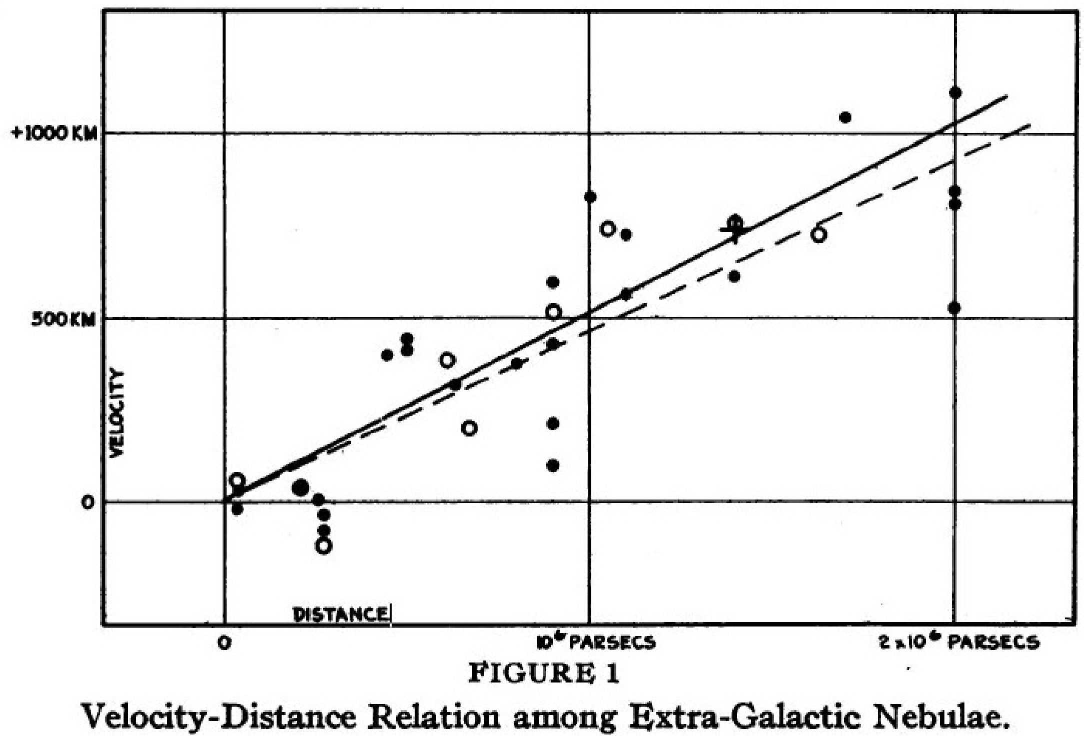

Figure 1: From Hubble's article[1].

1 Problem 1

1.1 Hubble's Discovery

Edwin Hubble published his results on cosmological expansion in a 1929 article[1]. His data are summarized in Fig. 1 and Table 1

1.2 Numerical Analysis

$$rK + X\cos\alpha\cos\delta + Y\sin\alpha\cos\delta + Z\sin\delta = v \tag{1}$$

Eq. 1 was simplified

$$v = a + br$$

Today we are going to reproduce the numerical analysis using python script with both the case of all data and the 9 groups method. As you can already see, the data has already been sorted into 9 groups based on the distance and the direction. The mechanism of the distribution is to first divide them into 3 groups with first the distance ranges from 0.032 to 0.5, the second from 0.5 to 1.0 and the third from 1.1 to 2.0. Then, according the their declinations, divide each group into 3 sub groups making totally 9 groups. Using the average distance and velocity of each group as shown in Table 2

Name	A(hours:minutes)	D(degrees:minutes)	r (Mpc)	v (km/s)	group
s.Mag,	..	..	0.032	+170	1
L.Mag.	..	..	0.0342	+290	1
N.G.C.6822	19:44.9	-14:48	0.2144	-130	2
598	1:33.9	+30:39	0.2634	-70	2
221	0:42.7	+40:52	0.2753	-185	2
224	0:42.7	+41:16	0.2755	-220	2
5457	14:03.2	+54:21	0.45	+200	3
4736	12:50.9	+41:07	0.5	+290	3
5194	13:29.9	+47:12	0.5	+270	6
4449	12:28.2	+44:06	0.63	+200	6
4214	12:15.6	+36:20	0.8	+300	5
3031	9:55.6	+69:04	0.9	-30	6
3627	11:20.2	+12:59	0.9	+650	4
4826	12:56.7	+21:41	0.9	+150	5
5236	13:37.0	-29:52	0.9	+500	5
1068	2:42.7	-00:01	1.0	+920	4
5055	13:15.8	+42:02	1.1	+450	9
7331	22:37.1	+34:25	1.1	+500	8
4258	12:19.0	+47:18	1.4	+500	9
4151	12:10.5	+39:24	1.7	+960	9
4382	12:25.4	+18:11	2.0	+500	8
4472	12:29.8	+08:00	2.0	+850	7
4486	12:30.8	+12:24	2.0	+800	7
4649	12:43.7	+11:33	2.0	+1090	7

Table 1: Hubble's data

1.3 Results

Using all the data, a=-40.78 ± 83.44 Km/s and b=454.16 ± 75.24 Km/s/Mpc, and the output plot is here 2

Using the 9 groups data, a=11.21 ± 126.59 Km/s and b=431.37 ± 116.58 Km/s/Mpc, and the output plot is here 3

Since the age of the universe is 1/Hubble's constant [3] , I have the age of the universe is $215 \pm 36 \times 10^7 yrs$ for all data method and $227 \pm 61 \times 10^7 yrs$ for nine groups method.

Group Number	Distance r (Mpc)	Radial velocity v (km/s)
1	0.033	+230
2	0.257	-151
3	0.475	+245
4	0.950	+785
5	0.866	+316
6	0.677	+146
7	2.000	+913
8	1.500	+500
9	1.400	+636

Table 2: Hubble's 9 groups data

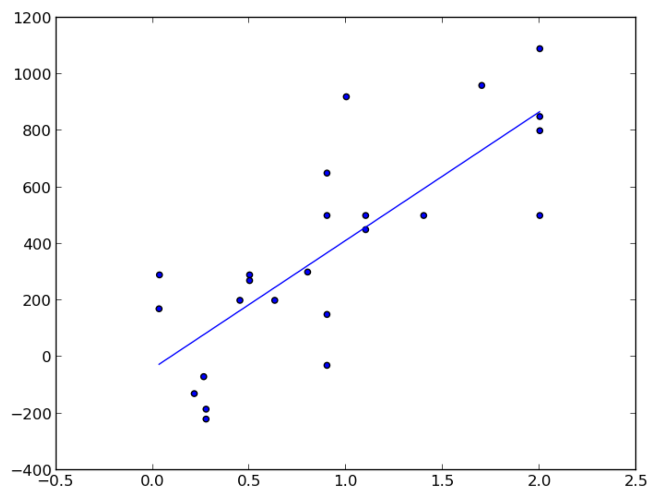

Figure 2: The output using all the data.

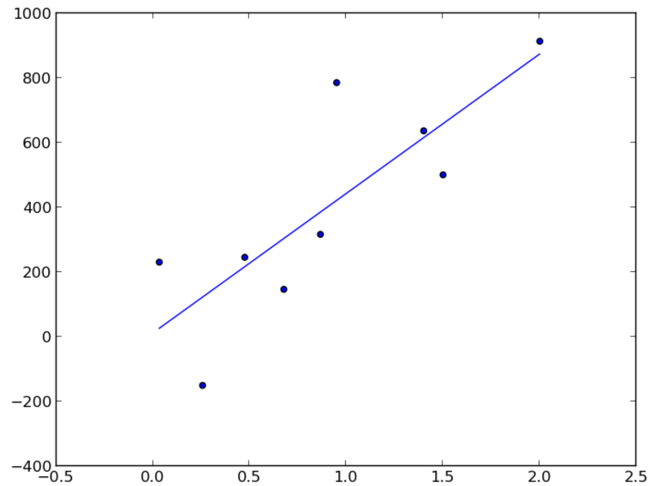

Figure 3: The result using 9 groups data.

2 Problem 2

2.1 Supernovae

Find Hubble's constant from the intercept and slope output of the supernova program and compare with Hubble's value. Explain any discrepancies you observe.

2.2 Numerical Analysis

As with the supernovae, it no longer obeys the simple non-relativistic form of the Hubble's law:$v = rK + const$, the relativistic form must be employed:

$$\mu = 25 + 5log_{10}(\frac{cz}{H_o}) + 1.086(1 - q_0)z + ...$$

Considering $q_0 = 0$ in our case, the equation can be written as:

$$\mu = 25 + 5log_{10}(\frac{c}{H_o}) + 5log_{10}z$$

Therefore we can calculate the H_o by means of the linear relationship between μ and $log_{10}z$,namely the intercept $I$and$H_o$can be linked by:

$$I = 25 + 5log_{10}(\frac{c}{H_o})$$

So$H_o = \frac{c}{10^{I/5-5}}$

2.3 result

The linear fits result is shown in picture 4. The slope is 5.553 ± 0.026 and the intercept is 44.152 ± 0.022 Correspondingly, the Hubble's constant is $H_o = 44.329km/s/Mpc$

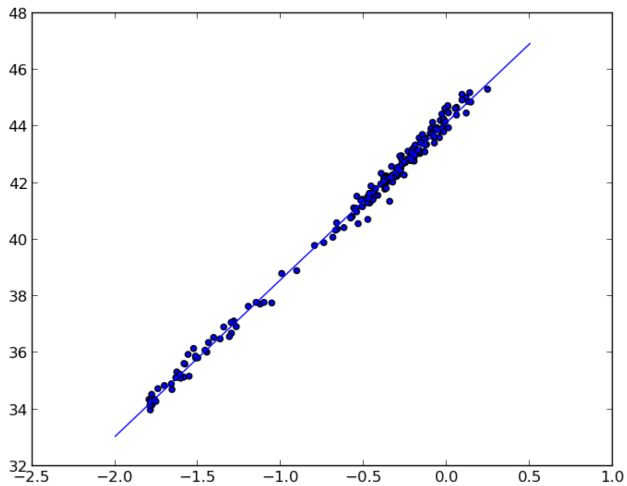

Figure 4: The output using supernovae using all the data.

3 Problem 3

3.1 LOW AND HIGH RED SHIFT

Divide the supernova data set into two subsets, low redshift and high redshift. Compute the slope separately for each of the two subsets. Can you conclude from your results that the expansion of the Universe is constant, accelerating, or decelerating?

3.2 Numerical analysis

Based on z values we can divide data into two groups, one with z smaller than 0.7 and those larger than that. To achieve that, I added some extra lines in the python script to write those two files.

3.3 result

For the group with high shift 5, we have The slope is 6.070 ± 0.098 and the intercept is 44.272 ± 0.031 Correspondingly, the Hubble's constant is $H_o = 41.952 km/s/Mpc$.

For the group with low shift 6, we have The slope is 5.346 ± 0.105 and the intercept is 43.875 ± 0.153. Correspondingly, the Hubble's constant is $H_o = 50.363 km/s/Mpc$.

Based on the result above, the expansion of the universe is increasing.

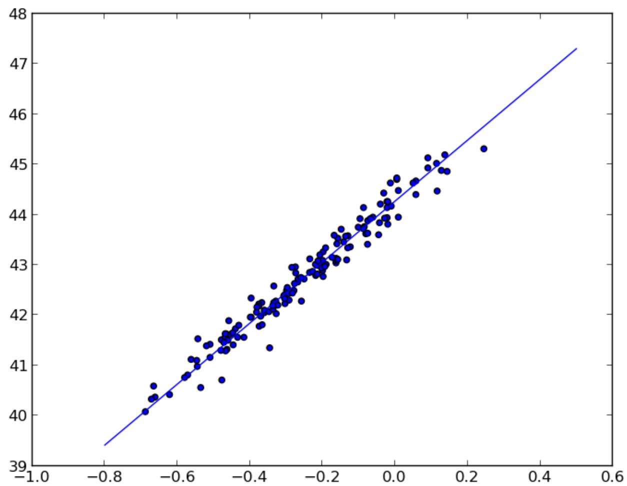

Figure 5: The output using supernovae using high shift the data.

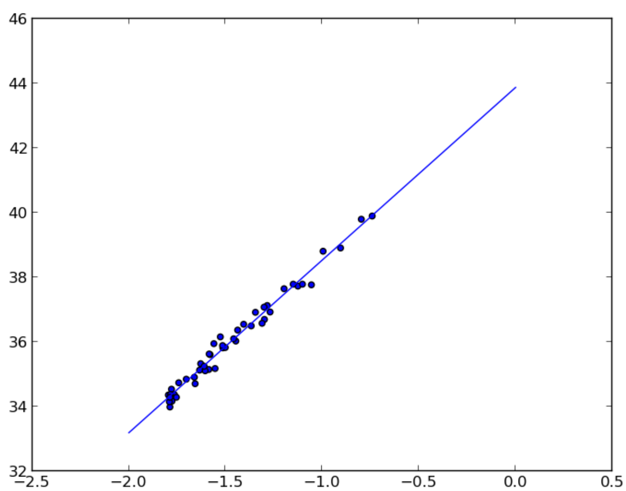

Figure 6: The output using supernovae using low shift the data.

Acknowledgements

I discussed this assignment with my classmates and used material from the cited references, but this writeup is my own.

References

[1] "A Relation Between Distance and Radial Velocity Among Extra-Galactic Nebulae", Edwin Hubble, Proc. Natl. Acad. Sci. USA **15**, 168-173 (1929).

[2] PHY 410-505 Webpage, `http://www.physics.buffalo.edu/phy410-505`.

[3] Wikipedia:Age of the universe,`http://en.wikipedia.org/wiki/Age_of_the_universe`.

A Appendix

A.1 python code

The following python code was used to obtain the results in this report:

```
#WH_PHY410_HW1_Hubble.py
import math
import matplotlib.pyplot as plt
import numpy as np
# distances in Mpc
r = [ 0.032, 0.034, 0.214, 0.263, 0.275, 0.275, 0.45, 0.5,
      0.5,   0.63,  0.8,   0.9,   0.9,   0.9,   0.9,  1.0,
      1.1,   1.1,   1.4,   1.7,   2.0,   2.0,   2.0,  2.0 ]

# velocities in km/s
v = [ +170, +290, -130, -70,  -185, -220, +200, +290,
      +270, +200, +300, -30,  +650, +150, +500, +920,
      +450, +500, +500, +960, +500, +850, +800, +1090 ]

n=len(r)                   #to read the number of sets of data

if n <= 2 :
       print 'Error! Need at least two data points!'
       exit()
#set needed quantities
s_x = 0
s_y = 0
s_xx = 0
s_xy = 0

for i in range(0,n):
     s_x=s_x+r[i]
     s_y=s_y+v[i]
     s_xx=s_xx+r[i]*r[i]
     s_xy=s_xy+r[i]*v[i]
#examine denominator
deno=(n*s_xx)-(s_x)**2
if deno<0.00001:
     print "wrong data, denominator is zeor when doing linear fits"
     exit()
a=(s_xx*s_y-s_x*s_xy)/deno
b=(n*s_xy-s_x*s_y)/deno
#uncertainty
sigma2=0
for i in range(0,n):
     sigma2=sigma2+(v[i]-(a+b*r[i]))**2
if n>2:
     sigma2=sigma2/(n-2)
```

```
    sigma=math.sqrt(sigma2)
    sigma_a=math.sqrt(sigma**2*s_xx/deno)
    sigma_b=math.sqrt(sigma**2*n/deno)
else:
    sigma=0
    sigma_a=0
    sigma_b=0
#plot commands
plt.scatter(r,v)
t=np.arange(r[0],r[n-1],0.001)
plt.plot(t,b*t+a)

# Print out results
print ' Least squares fit of', n, 'data points'
print ' -----------------------------------------'
print " Hubble's constant slope   b = {0:6.2f} +- {1:6.2f}  km/s/Mpc".format( b, si
print " Intercept with r axis     a = {0:6.2f} +- {1:6.2f}  km/s".format( a, sigma_
print ' Estimated v error bar sigma =', round(sigma, 1), 'km/s'

plt.show()

#PHY410 homework1 supernovae
import math
import numpy as np
import matplotlib.pyplot as plt


def chi_square_fit(x, y, err):
    n = len(x)
    if n < 2 :
        print 'Error! Need at least 2 data points!'
        exit()
    S = sum(1/err[i]**2 for i in range(n))
    if abs(S) < 0.00001 :
        print 'Error! Denominator S is too small!'
        exit()
    S_x = sum(x[i]/err[i]**2 for i in range(n))
    S_y = sum(y[i]/err[i]**2 for i in range(n))
    t = [(x[i] - S_x/S) / err[i] for i in range(n)]
    S_tt = sum(t_i**2 for t_i in t)
    if abs(S_tt) < 0.00001 :
        print 'Error! Denominator S is too small!'
        exit()
    b = sum(t[i]*y[i]/err[i] for i in range(n)) / S_tt
    a = (S_y - S_x * b) / S
    sigma_a2 = (1 + S_x**2/S/S_tt) / S
    sigma_b2 = 1/S_tt
    if sigma_a2 < 0.0 or sigma_b2 < 0.0 :
```

```
        print 'Error! About to pass a negative to sqrt'
        exit()
    sigma_a = math.sqrt(sigma_a2)
    sigma_b = math.sqrt(sigma_b2)
    chi_square = sum(((y[i] - a - b*x[i]) / err[i])**2 for i in range(n))
    return(a, b, sigma_a, sigma_b, chi_square)

print ' Chi-square fit of supernova data to a straight line'
print ' Reference: http://dark.dark-cosmology.dk/~tamarad/SN/'
data_file = open('Davis07_R07_WV07.dat', 'r')
#data_file = open('error1.dat', 'r')
#data_file = open('error2.dat', 'r')
#data_file = open('error3.dat', 'r')
#data_file = open('error4.dat', 'r')
lines = data_file.readlines()
data_file.close()
data = [ str.split(line) for line in lines if line[0] != ';' ]
print ' read', len(data), 'data values'
#divide data into two files:
fg=open('supernovagt07.dat','w')
fs=open('supernovast07.dat','w')
for datum in data:
    if (math.log10(float(datum[1]))>-0.7):
        fg.write(datum[0]+' '+datum[1]+' '+datum[2]+' '+datum[3]+'\n')
    else:
        fs.write(datum[0]+' '+datum[1]+' '+datum[2]+' '+datum[3]+'\n')
fg.close()
fs.close()
#end
logz_data = [math.log10(float(datum[1])) for datum in data]
mu_data = [float(datum[2]) for datum in data]
mu_err_data = [float(datum[3]) for datum in data]


for i_mu_err_data in mu_err_data :
    if abs(i_mu_err_data) < 0.000001 :
        print 'Error! Uncertainties are too small!'
        exit()

plt.scatter( logz_data, mu_data )


fit = chi_square_fit(logz_data, mu_data, mu_err_data)
print ' slope =', fit[1], ' +- ', fit[3]
print ' intercept =', fit[0], '+-', fit[2]
if len(data) - 2 > 0 :
    print ' chi-square/d.o.f. = ', fit[4]/(len(data)-2)
else :
```

```
    print ' chi-square/d.o.f. undefined'
hubble=(3*10**8)/10**(fit[0]/5.-5.0)
print hubble
t=np.arange(-2.0,0.5,0.001)
plt.plot(t,fit[1]*t+fit[0])

plt.show()
```