Fitting Data To models

EE16B014
G.HARSHA KANAKA ESWAR

April 07,2018

Abstract

In this report we are going to discuss about fitting of data by taking example of Bessel function and approximate it to two function:1)$Acos(x_i)+Bsin(x_i)$ and 2)$A\frac{cos(x_i)}{\sqrt{x_i}}+B\frac{sin(x_i)}{\sqrt{x_i}}$ which are near functions to Bessel function,and we will evaluate the their coefficients by best fitting method and we will see the effect of noise in the fitting.

1 Bessel Function:

Bessel function is generally the a solution to the differential equation:

$$x^2\frac{d^2y}{dx^2}+x\frac{dy}{dx}+(x^2-\alpha^2)y=0$$

- The function is in such away that the output is same for both $+\alpha$ and $-\alpha$ and its a smooth function with α.
- The most important cases are when α is an integer or half-integer. Bessel functions for integer α are also known as cylinder functions or the cylindrical harmonics because they appear in the solution to Laplace's equation in cylindrical coordinates. Spherical Bessel functions with half-integer α are obtained when the Helmholtz equation is solved in spherical coordinates.

Lets take α as v and proceed.Now we have approximate form for the Bessel function as given below:

$$J_v(x)\approx\sqrt{\frac{2}{\pi x}}cos\left(x-\frac{v\pi}{2}-\frac{\pi}{4}\right)$$

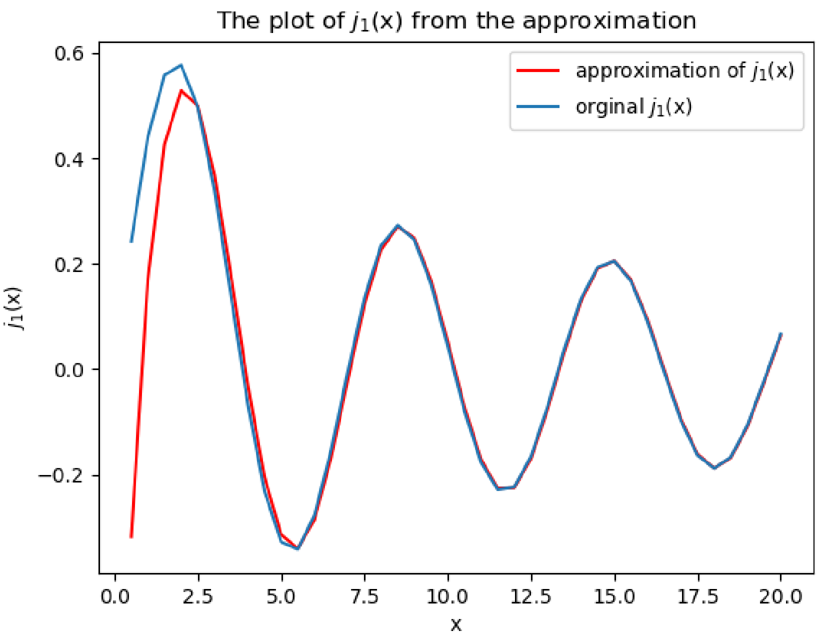
Now we first plot the approximation of Bessel function along with the original one.

```
from numpy import *
from pylab import *
from math import *
from scipy import *
from scipy.special import j1
x=linspace(0,20,41)
x=x[1:]

jk=((2/pi)**(1/2))*cos(x-(3*pi/4))/(x**(1/2.0))

plot(x,jk,"r",label="approximation of $j_1$(x)")
plot(x,j1(x),label="orginal $j_1$(x)")
legend(loc="upper right")
xlabel('x')
ylabel('$j_1$(x)')
title('The plot of $j_1$(x) from the approximation')
show()
```

Plot:



As we can see the approximation graph is slightly distorted from the original one which means our approximate function is not that accurate for less values of x.

Now lets define a function called *calcnu* Which can calculate the coefficients of A,B for our required functions when approximated as shown below:

$$A cos(x_i) + B sin(x_i) \approx J_1(x_i)$$
$$A\frac{cos(x_i)}{\sqrt{x_i}} + B\frac{sin(x_i)}{\sqrt{x_i}} \approx J_1(x_i)$$

Calcnu:

Here is the function which we have defined:

```
calcnu(y,xo,k,eps,model)
```

It takes the input arguments y,which defines the vector range for our approximation;$x_0$defines from which point of vector y to begin with,k is the Bessel function v value,eps

is the error to be added to the function ,model defines the approximation to be considered whether to take the first or the second one.

```
def calcnu(y,xo,k,eps,model):
        l=where(y>=xo)
        y1=y[l]
        l1=len(y1)
        A=zeros((l1,2))
        b=j1(y1)+eps*(rand(len(y1)))
        if model==0:
                A[:,0]=cos(y1)
                A[:,1]=sin(y1)
        if model==1:
                A[:,0]=cos(y1)/(y1**(0.5))
                A[:,1]=sin(y1)/(y1**(0.5))
        c=lstsq(A,b)[0]
        i1=arccos(c[0]/((c[0])**2+(c[1])**2)**(0.5))
        v1=-1*(-2*i1/pi+0.5)
        return v1
```

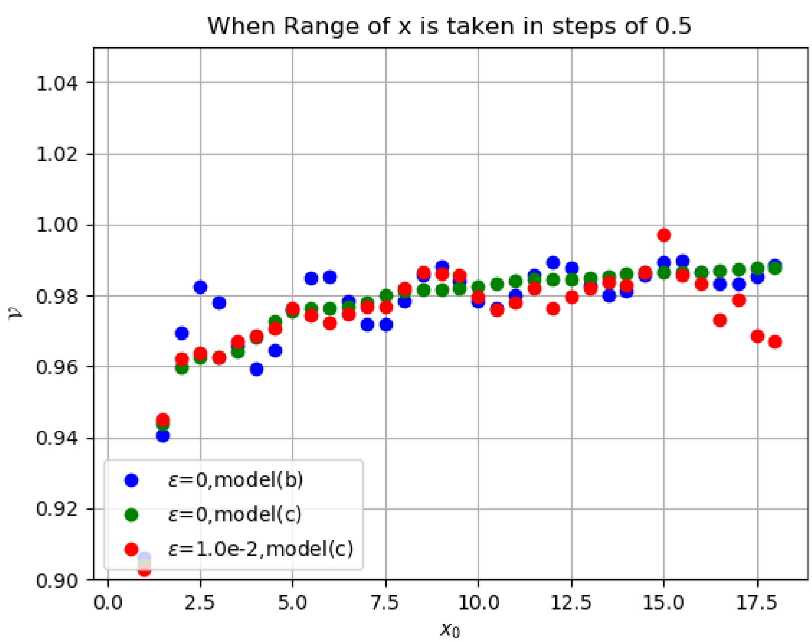
It returns the value of v (v is the subscript of$J_v(x)$),So that we can know that the approximation is very good if v value is very near to *one*.(Because we have considered the function to be $J_1(x)$.
The function is basically calculating coefficients then getting the angle ϕ when our function is written in the form $\sqrt{A^2+B^2}cos(x_i+\phi)$ where ϕ is nothing but $cos^{-1}\left(\frac{A}{\sqrt{A^2+B^2}}\right)$ so that we can get v value from the approximate Bessel function $\sqrt{\frac{2}{\pi x}}cos\left(x-\frac{v\pi}{2}-\frac{\pi}{4}\right)$.By equating $\phi=\frac{v\pi}{2}+\frac{\pi}{4}$.

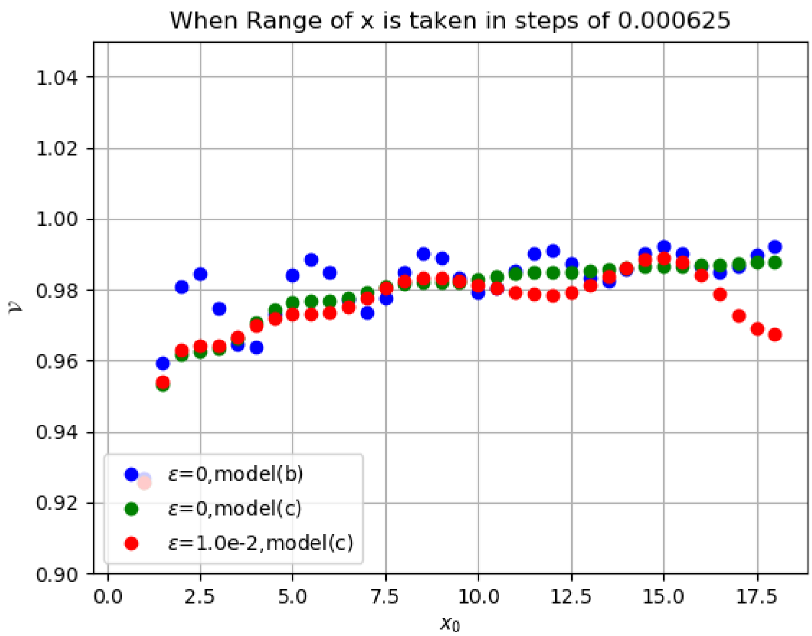
Calling the calcnu function for the v values and plotting:

```
y=arange(0.5,20.5,0.000625)
for xo in y3:
        nu=calcnu(y,xo,0,0,0)
        v1.append(nu)
for xo in y3:
        nu=calcnu(y,xo,0,0,1)
        v2.append(nu)
for xo in y3:
        nu=calcnu(y,xo,0,0.01,1)
        v3.append(nu)
```

```
#Plots
fig1=figure(2)
plot(y3,v1,'bo',label="$\epsilon$=0,model(b)")
plot(y3,v2,'go',label="$\epsilon$=0,model(c)")
plot(y3,v3,'ro',label="$\epsilon$=1.0e-2,model(c)")
title("Range of x in steps of 0.000625")
xlabel("$x_0$")
ylabel("$\mathcal{V}$")
grid()
axes=plt.gca()
axes.set_ylim([0.9,1.05])
legend(loc="lower left")
show()
```

Graph of v values for both approximate functions :





Conclusions:

- Looking at the Plots v vs x_0 we can say that model(c) is more accurate than model(b).
- By Varying the no of measurements by not changing the steps taken for *range of x* change from 0.5 to 0.000625we can say that the effect of noise in the graph for model(c) is considerably less.
- Moreover quality fit of error is very poor in the end values of x_o because of only less values are given for leastsq approximation.
- There is also a large deviation from "v" value "one" in the beginning,this is because the Bessel function is more near to the $\sqrt{\frac{2}{\pi x}} cos\left(x - \frac{v\pi}{2} - \frac{\pi}{4}\right)$ approximation from higher values of "x" so the approximate value is not near to one in the beginning.