GA_g_k

February 9, 2021

[1]:

```
%pylab inline
import numpy as np
import matplotlib.pyplot as plt
from sko.GA import GA
import warnings
import xlrd
warnings.filterwarnings('ignore')
```

```
Populating the interactive namespace from numpy and matplotlib
```

[2]:

```
xl=xlrd.open_workbook(r'C:\Users\29691\Documents\WeChat␣
 ↪Files\wxid_xnxilu4ixtb211\FileStorage\File\2021-02\Fungi_moisture_curves(1).
 ↪xlsx')
```

[3]:

```
def f_fun(t,a,b,c,d):
    y=[0.0]*len(t);
    for i in range(len(t)):
        if (t[i]<=d):
            y[i]=a*exp(-1/b*(d-t[i])**2)
        else:
            y[i]=a*exp(-1/c*(t[i]-d)**2)
    return y
```

[4]:

```
def obj_fun(p):
    global x_true
    global y_true
    a,b,c,d=p
    res=np.square(f_fun(x_true,a,b,c,d)-y_true).sum()
    return res
```

[16]:

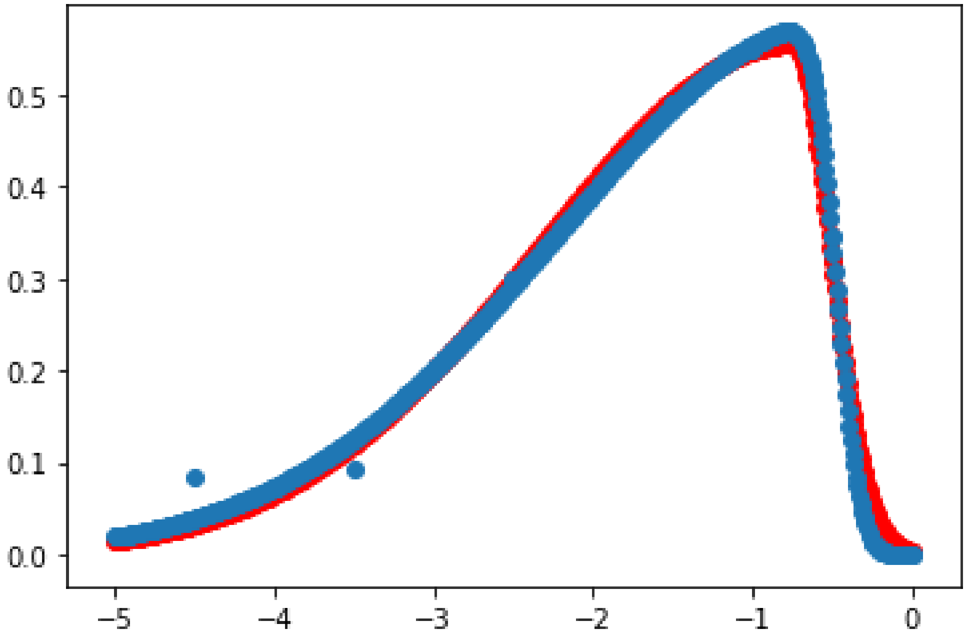
```
def printf(i):
    table=xl.sheets()[i]
    global x_true
    global y_true
    x_true=np.array(table.col_values(1,1))
    y_true=np.array(table.col_values(2,1))
    plot(x_true,y_true,'o')
```

[20]:

```
def GA_g(i,aa,l,u):
    table=xl.sheets()[i]
    global x_true
    global y_true
    x_true=np.array(table.col_values(1,1))
    y_true=np.array(table.col_values(2,1))
    ga = GA(func=obj_fun,n_dim=4,size_pop=50,max_iter=500,lb=[0,0,0,l],␣
 ↪ub=[aa,100,100,u])
    best_params,residuals=ga.run()
    y_predict=f_fun(x_true,*best_params)
    fig,ax=plt.subplots()
    ax.plot(x_true,y_true,'o')
    ax.scatter(x_true,y_predict,c='r')
    plt.show()
    return best_params,residuals
```

[21]:

```
GA_g(1,1,-2,0)
```

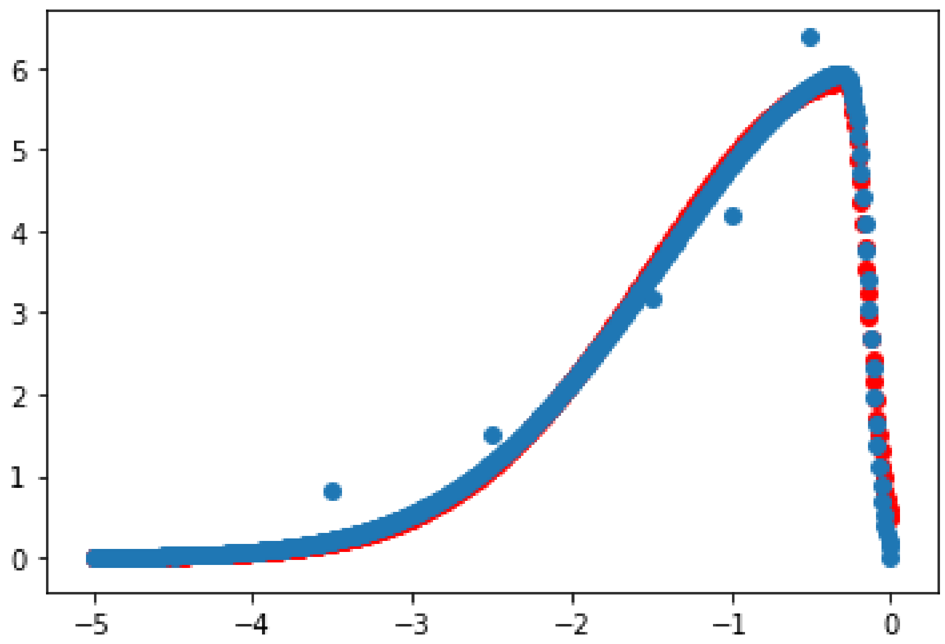


[21]:

```
(array([ 0.55429766,  4.97956714,  0.10852851, -0.74434539]),
 array([0.09921496]))
```

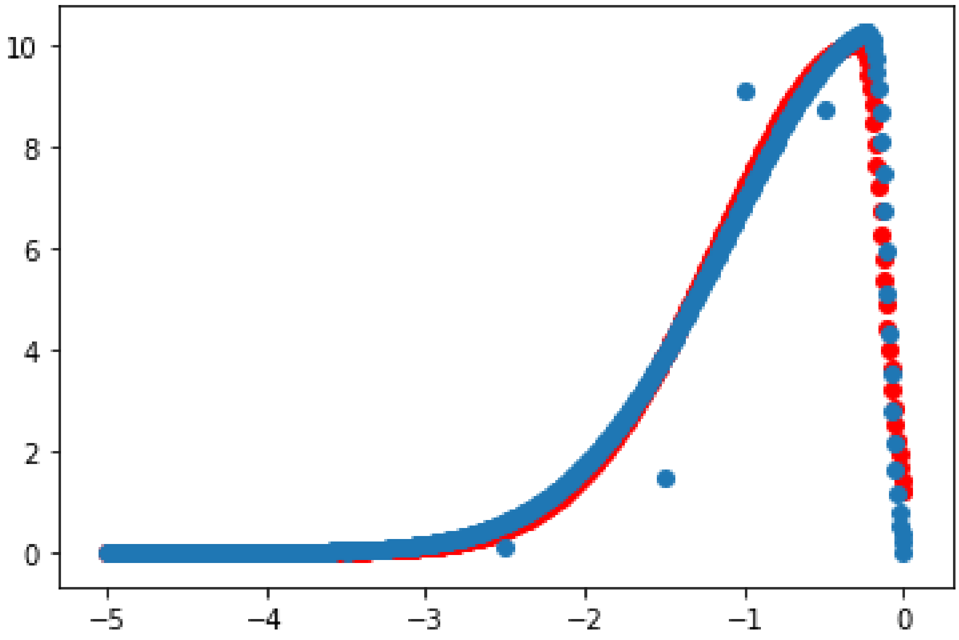
[22]:

```
GA_g(2,7,-1,1)
```



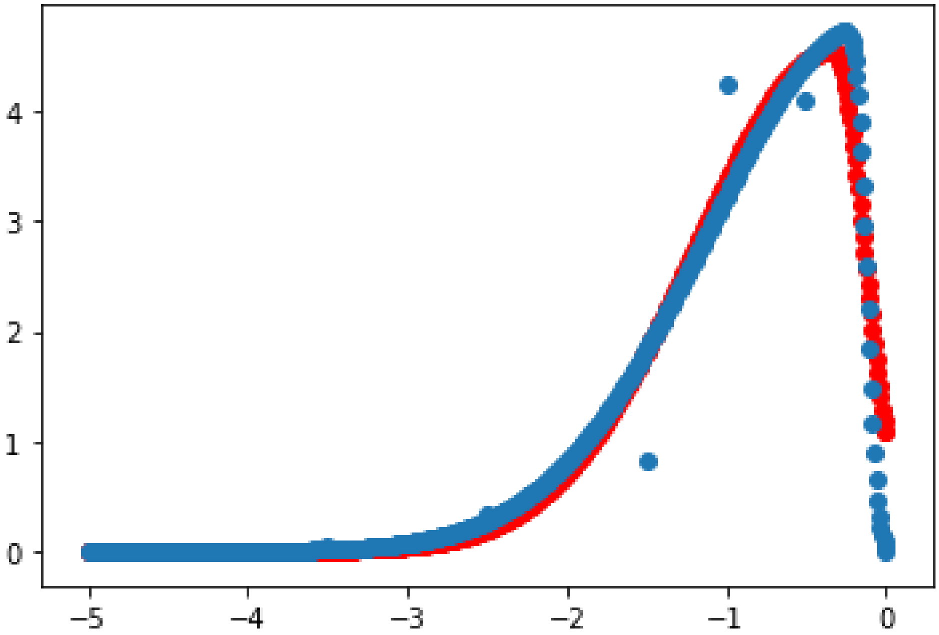
```
[22]: (array([ 5.8282527 ,  2.98481537,  0.03055166, -0.27392564]),
        array([5.23522102]))
```

```
[24]: GA_g(3,12,-1,0)
```



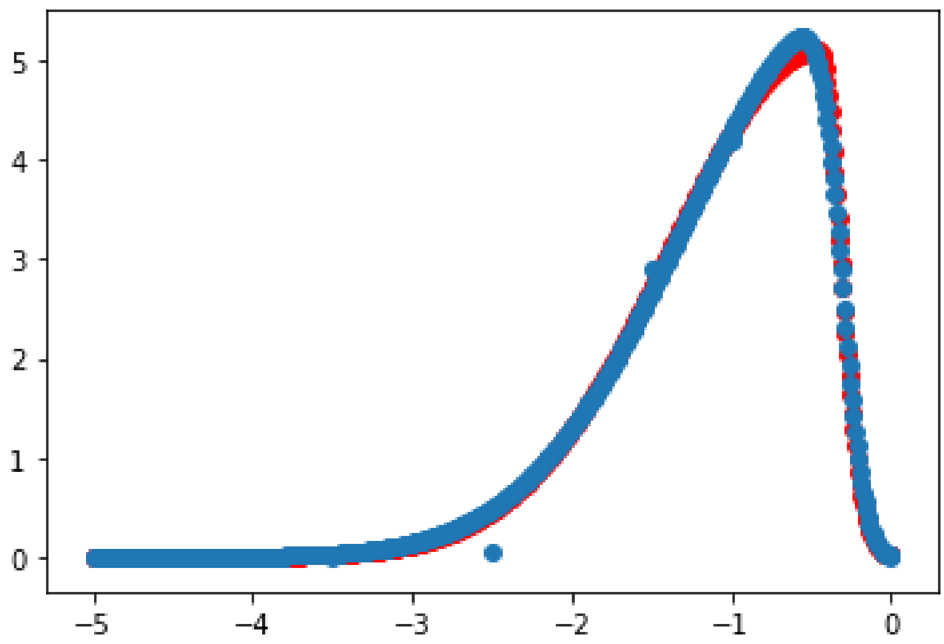
```
[24]: (array([10.03501697,  1.60670942,  0.03364133, -0.26559647]),
       array([55.68667338]))
```

```
[27]: GA_g(4,5,-1,0)
```



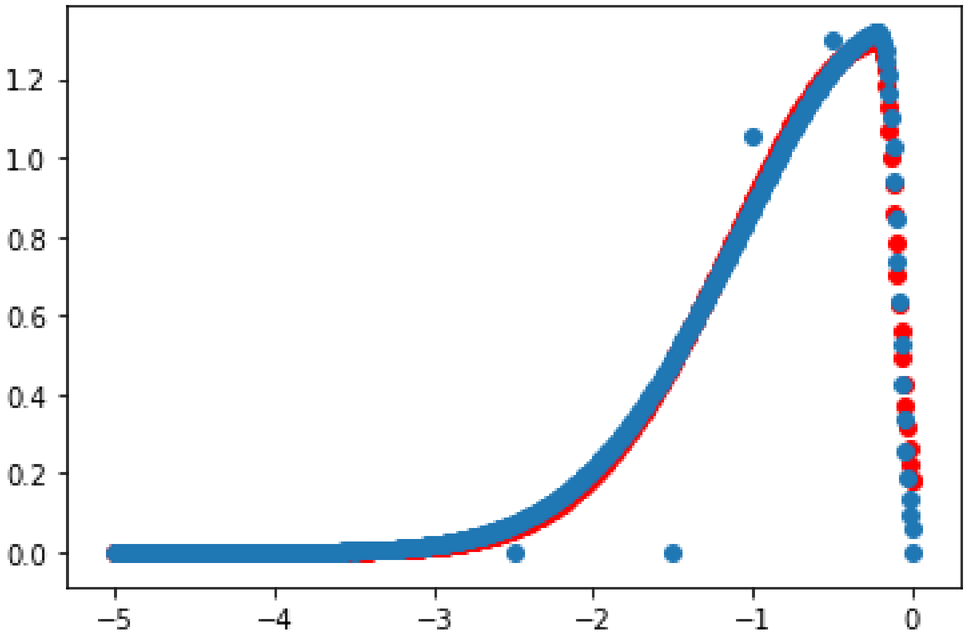
```
[27]: (array([ 4.55094389,  1.50985588,  0.07325001, -0.3239673 ]),
       array([23.50420292]))
```

```
[29]: GA_g(5,6,-2,0)
```



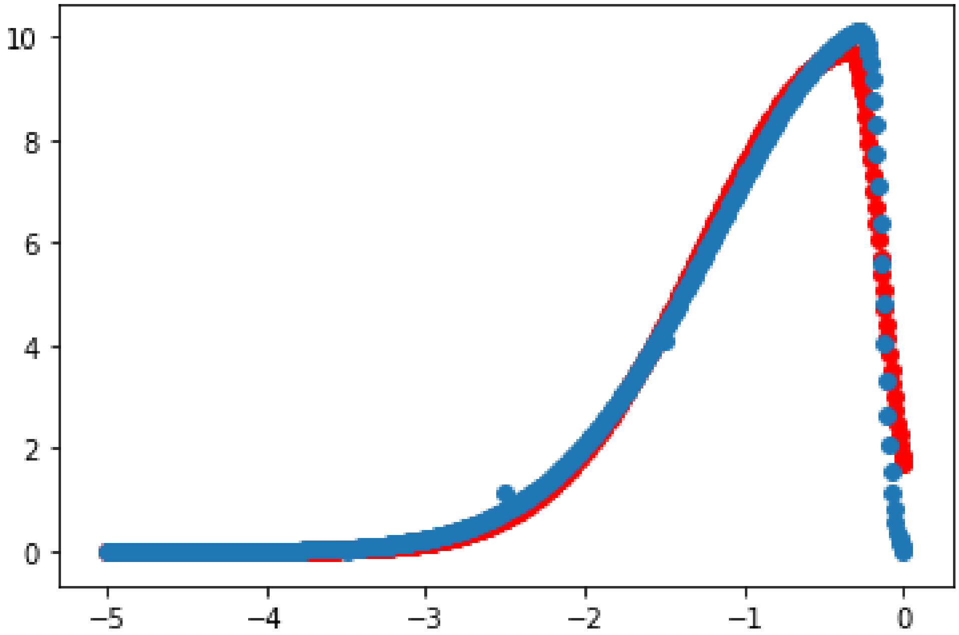
```
[29]: (array([ 5.09610351,  1.75323207,  0.03385497, -0.44601841]),
        array([2.41306318]))
```

```
[34]: GA_g(6,1.4,-2,0)
```



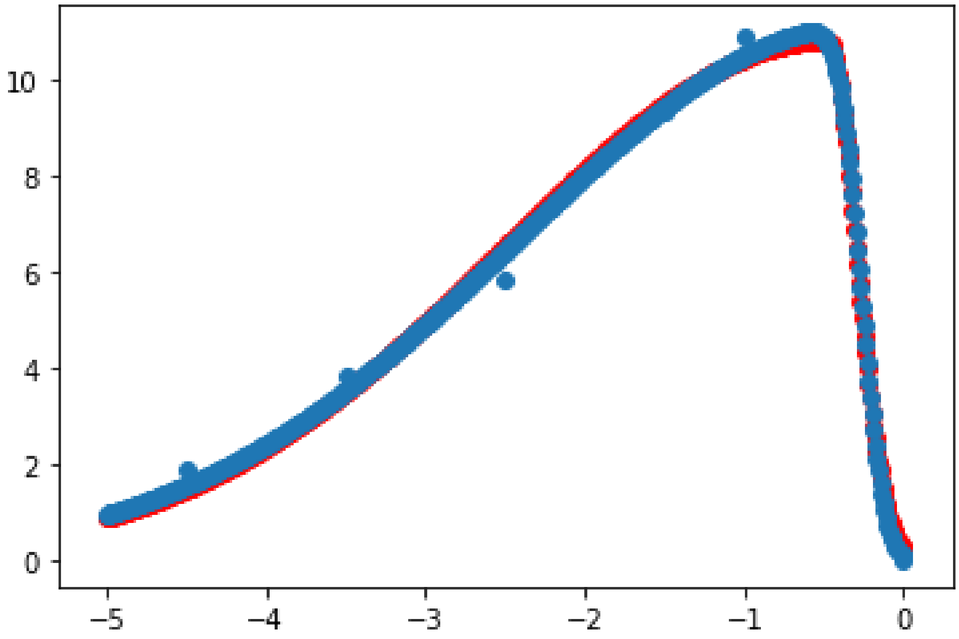
```
[34]: (array([ 1.29464653,  1.70293031,  0.02139164, -0.20384336]),
       array([0.49358759]))
```

```
[33]: GA_g(7,11,-1,0)
```



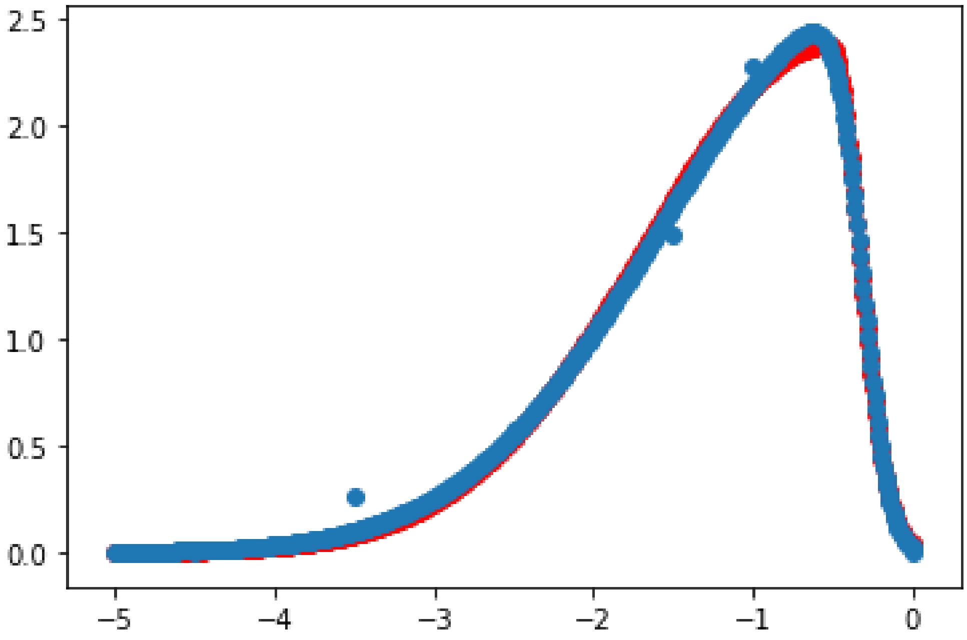
```
[33]: (array([ 9.71942219,  1.72630623,  0.06392375, -0.33408954]),
       array([83.18315957]))
```

```
[36]: GA_g(8,12,-2,0)
```



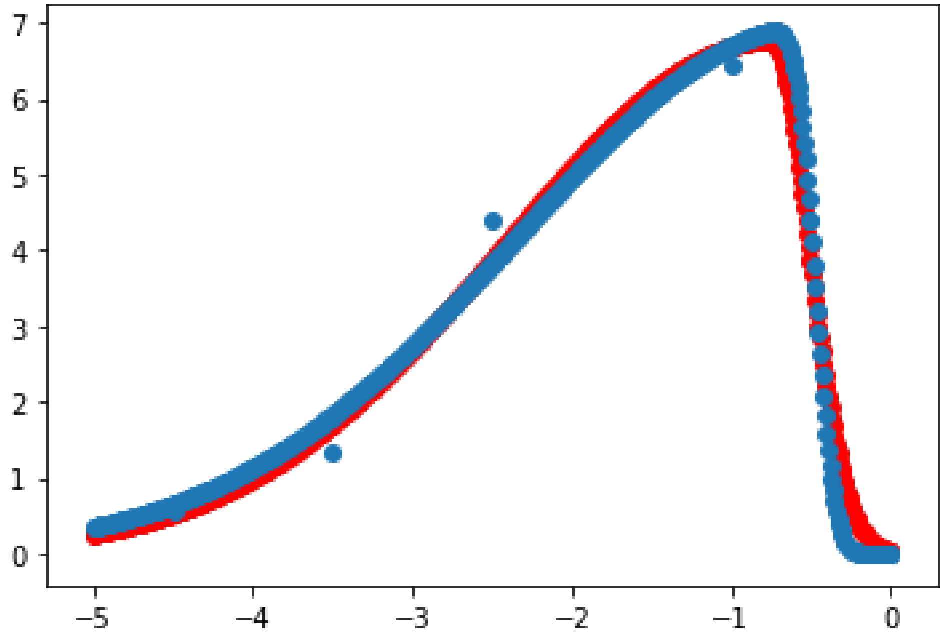
```
[36]: (array([10.78834649,  8.14199029,  0.05930578, -0.47292258]),
       array([6.25215103]))
```

```
[38]: GA_g(9,3,-1.5,0)
```



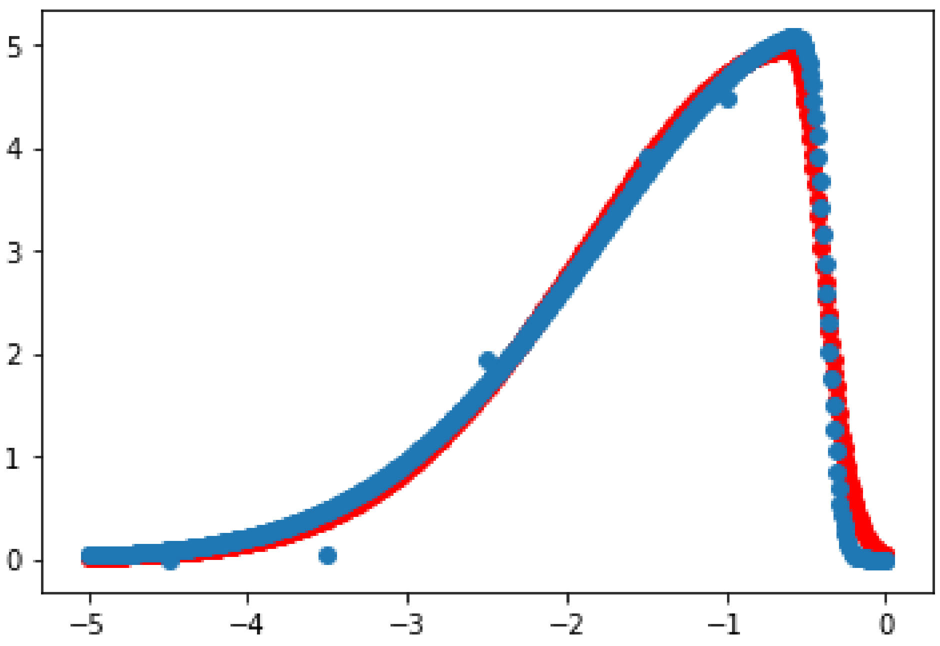
```
[38]: (array([ 2.37485529,  2.66153189,  0.06341925, -0.5167926 ]),
        array([0.36371574]))
```

```
[41]: GA_g(10,8,-1,0)
```



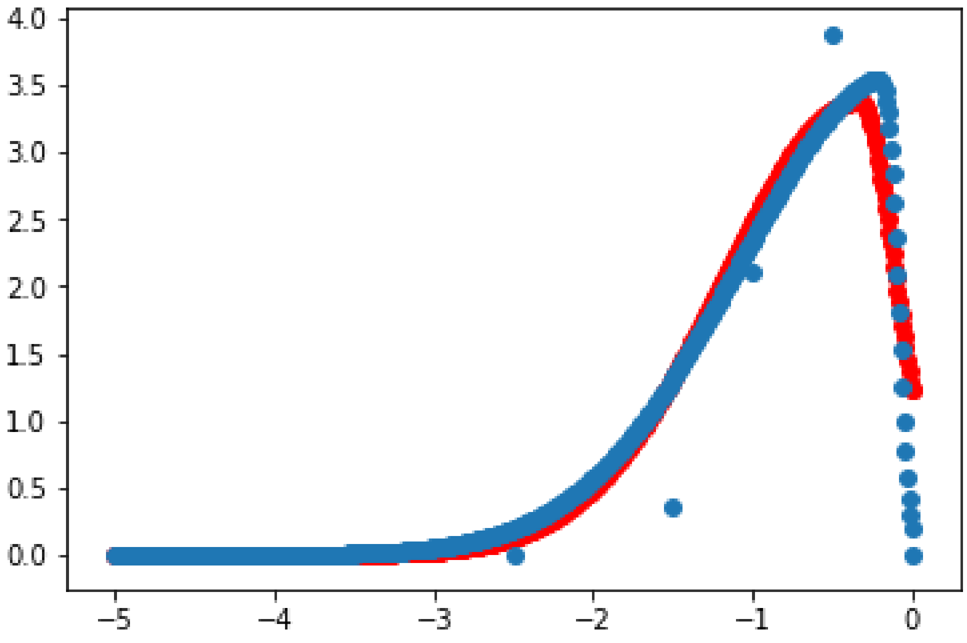
```
[41]: (array([ 6.76411687,  5.54435039,  0.10428103, -0.74129604]),
        array([27.58014705]))
```

```
[43]: GA_g(11,6,-1,0)
```



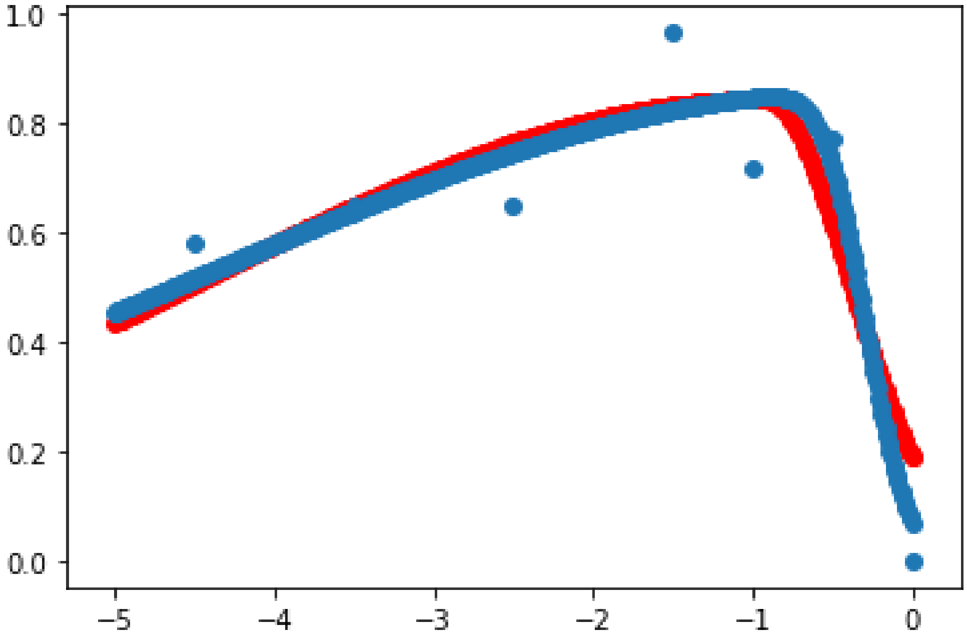
```
[43]: (array([ 4.95814229,  3.36268777,  0.07011034, -0.58654622]),
        array([14.08928216]))
```

```
[45]: GA_g(12,4,-1,0)
```



```
[45]: (array([ 3.38443237,  1.44936992,  0.10793805, -0.33107867]),
       array([16.19753424]))
```

```
[47]: GA_g(13,1,-1.5,0)
```



```
[47]: (array([ 0.84497057, 25.15396432,  0.56318035, -0.91993901]),
       array([0.68253886]))
```