黄玟瑜 19335074 huangmy73@mail2.sysu.edu.cn

Homework 10 数值计算方法, 2021 春

2021-05-21

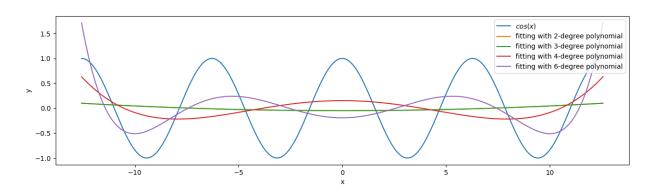
分别用 2,3,4,6 阶多项式拟合函数 y = cos(x), 并将拟合曲线与函数曲线 y = cos(x) 进行比较。

编写代码如下:

```
import matplotlib.pyplot as plt
   import numpy as np
   # 多项式拟合函数y=f(x), k: 多项式的阶数
   def Polyfit(x:np.array, y:np.array, k):
      cols = len(y)
                           # 获取数据的个数
6
      A = np.zeros((k+1,cols))
                             # 初始化A
      for i in range(k+1):
         for j in range(cols):
9
             A[i, j] = pow(x[j], k-i)
                                     # A[i, j] = xj^{(k-i)}, A对应求解矛盾方程组的A的转置
10
      y = A.dot(y)
                   # b = AT * y
11
12
      A = A.dot(A.T) # A = AT * A
      x = np.linalg.solve(A, y) # 解方程组
13
14
15
      return x
16
17
   plt.figure(figsize=(16,4))
18
   X=np.linspace(-4*np.pi,4*np.pi,1024,endpoint=True)# [-4 , 4 ]的1024个值
19
   C=np.cos(X)
20
   plt.plot(X,C, label="$cos(x)$")
21
22
  a=Polyfit(X,C,2)
                    #用2次多项式拟合x, y数组
                    #拟合完之后用这个函数来生成多项式对象
  b=np.poly1d(a)
24
                    #生成多项式对象之后,就是获取2在这个多项式处的值
25
   c=b(X)
26
   plt.plot(X,c, label="fitting with 2-degree polynomial")
27
   a=Polyfit(X,C,3)
                    #用3次多项式拟合x, y数组
28
                    #拟合完之后用这个函数来生成多项式对象
  b=np.poly1d(a)
29
   c=b(X)
                    #生成多项式对象之后,就是获取2在这个多项式处的值
30
   plt.plot(X,c, label="fitting with 3-degree polynomial")
31
32
  a=Polyfit(X,C,4)
                    #用4次多项式拟合x, y数组
33
                    #拟合完之后用这个函数来生成多项式对象
  b=np.poly1d(a)
34
   c=b(X)
                    #生成多项式对象之后,就是获取2在这个多项式处的值
35
   plt.plot(X,c, label="fitting with 4-degree polynomial")
36
37
  a=Polyfit(X,C,6)
                    #用6次多项式拟合x, y数组
38
                    #拟合完之后用这个函数来生成多项式对象
  b=np.poly1d(a)
39
                    #生成多项式对象之后,就是获取2在这个多项式处的值
   c=b(X)
40
   plt.plot(X,c, label="fitting with 6-degree polynomial")
41
42
43
44
  a=Polyfit(X,C,10)
                    #用10次多项式拟合x, y数组
                    #拟合完之后用这个函数来生成多项式对象
b=np.poly1d(a)
                    #生成多项式对象之后, 就是获取x在这个多项式处的值
   c=b(X)
  plt.plot(X,c, label="fitting with 10-degree polynomial")
```

```
48
49
   a = Polyfit(X, C, 14)
                    #用14次多项式拟合x, y数组
   b=np.poly1d(a)
                    #拟合完之后用这个函数来生成多项式对象
50
   c=b(X)
                    #生成多项式对象之后,就是获取2在这个多项式处的值
51
   plt.plot(X,c, label="fitting with 14-degree polynomial")
52
53
  plt.ylabel("y")
55
  plt.xlabel("x")
56
57
   plt.legend()
   plt.show()
```

结果如下所示:



继续用 10、14 阶多项式来拟合,可以看到拟合函数越来越接近原函数,事实上当多项式阶数为 16 时肉眼观察到拟合曲线和原曲线已完全重合。

