f-16-jupyter-hoejere-grad

March 25, 2021

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
```

[2]:
```
x = np.array([-2.1, -1.9, -1.5, -0.8, -0.3, 0.1, 0.5, 1.2, 1.3, 1.7, 2.4])
y = np.array([ 0.1,  0.4, -0.1, -0.6, -0.5, 0.1, 1.0, 1.3, 0.7, 0.1, 0.2])
```

[3]:
```
cols = len(x)
cols
```

[3]:
```
11
```

[4]:
```
a = np.vander(x, cols)
```

[5]:
```
koeffs = np.linalg.solve(a, y[:, np.newaxis])
```

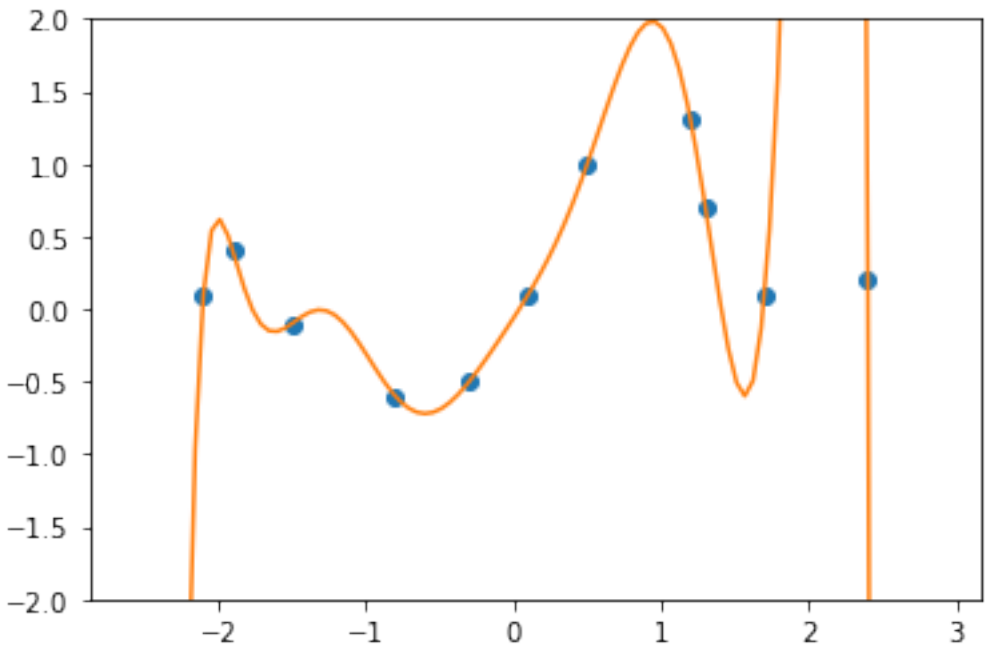
[6]:
```
koeffs
```

[6]:
```
array([[-0.09319931],
       [-0.06477183],
       [ 0.94400805],
       [ 0.72021707],
       [-2.85405456],
       [-2.04987989],
       [ 2.47820931],
       [ 0.95035073],
       [ 0.40938982],
       [ 1.57199016],
       [-0.06246781]])
```

[7]:
```
t = np.linspace(x.min()-0.5, x.max()+0.5, 100)

fig, ax = plt.subplots()
ax.set_ylim(-2.0, 2.0)
ax.plot(x, y, 'o')
ax.plot(t, np.vander(t, cols) @ koeffs)
```

[7]:
```
[<matplotlib.lines.Line2D at 0x120d761c0>]
```



[8]:
```
def forbedret_gram_schmidt(a):
    _, k = a.shape
    q = np.copy(a)
    r = np.zeros((k, k))
    for i in range(k):
        r[i, i] = np.linalg.norm(q[:, i])
        q[:, i] /= r[i, i]
        r[[i], i+1:] = q[:, [i]].T @ q[:, i+1:]
        q[:, i+1:] -= q[:, [i]] @ r[[i], i+1:]
    return q, r
```

[9]:
```
cols = 7
cols
```

[9]: 7

[10]:
```
a = np.vander(x, cols)
```

[11]:
```
q, r = forbedret_gram_schmidt(a)
```

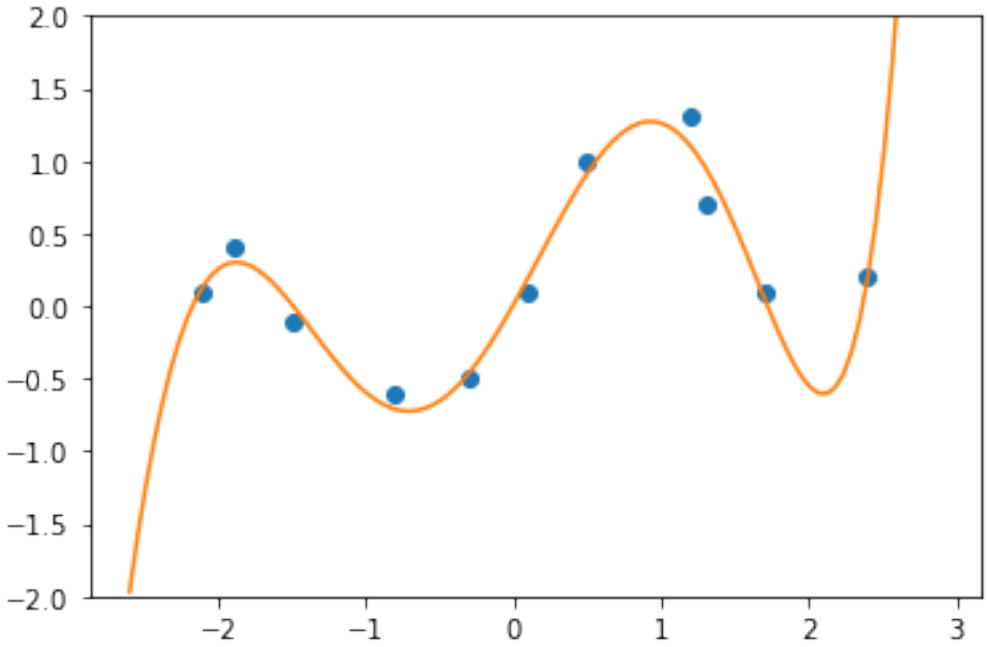
[12]:
```
c = q.T @ y[:, np.newaxis]
```

[13]:
```
koeffs = np.linalg.solve(r, c)
```

[14]:
```
t = np.linspace(x.min()-0.5, x.max()+0.5, 100)

fig, ax = plt.subplots()
ax.set_ylim(-2.0, 2.0)
ax.plot(x, y, 'o')
ax.plot(t, np.vander(t, cols) @ koeffs)
```

[14]:
```
[<matplotlib.lines.Line2D at 0x120e5e370>]
```



[15]:
```
rest = y[:, np.newaxis] - np.vander(x, cols) @ koeffs
np.linalg.norm(rest)
```

[15]:
```
0.400837595209125
```

[16]:
```
rest
```

[16]:
```
array([[-0.04036569],
       [ 0.09963581],
       [-0.11044335],
       [ 0.11108815],
       [-0.03960429],
       [-0.09563144],
       [ 0.08603725],
       [ 0.19850296],
       [-0.25445747],
```

```
       [ 0.04795903],
       [-0.00272098]])
```

[17]:
```
np.linalg.norm(rest)/cols
```

[17]:
```
0.05726251360130357
```

[18]:
```
a = np.vander(x, cols)
a.shape
```

[18]:
```
(11, 7)
```

[19]:
```
u, s, vt = np.linalg.svd(a, full_matrices=False)
s
```

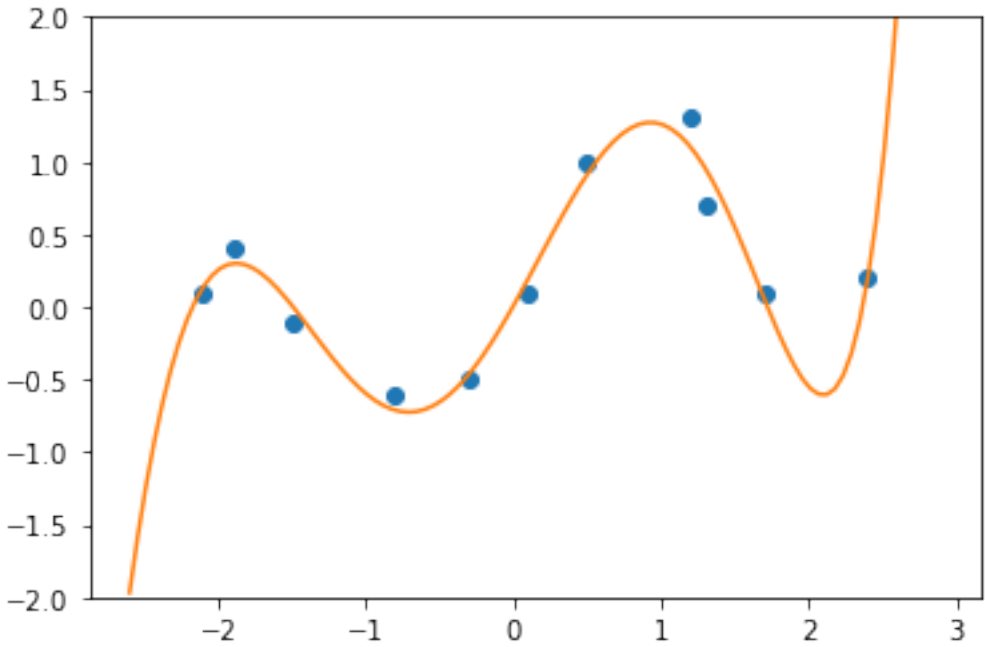
[19]:
```
array([226.92782643,  80.05387166,   7.40000455,   3.66969287,
         1.98866274,   0.71972993,   0.50406661])
```

[20]:
```
koeffs_svd = vt.T @ (np.diag(1/s) @ (u.T @ y[:, np.newaxis]))
```

[21]:
```
t = np.linspace(x.min()-0.5, x.max()+0.5, 100)

fig, ax = plt.subplots()
ax.set_ylim(-2.0, 2.0)
ax.plot(x, y, 'o')
ax.plot(t, np.vander(t, cols) @ koeffs_svd)
```

[21]:
```
[<matplotlib.lines.Line2D at 0x120ebbfa0>]
```



[22]:
```
rest_svd = y[:, np.newaxis] - np.vander(x, cols) @ koeffs_svd
np.linalg.norm(rest_svd)
```

[22]:
```
0.4008375952091247
```

[23]:
```
np.linalg.norm(rest_svd - rest)
```

[23]:
```
1.9638163267674327e-14
```

[ ]:
```

```