assignment1_q3

March 7, 2025

[29]:
```python
# (a)
import numpy as np
np.random.seed(123)

def getData(n):
    # annotation: the wrong version
    mu = np.array([-2.0, 3.0]).reshape((2, 1)) # mu = np.array([-2.0, 3.0])
    A = np.array([[1.0, 2.0], [3.0, 4.0]])
    X = np.random.normal(size= n*2)
    X = X.reshape((2, n))
    X = A.dot(X) + mu # X = (X + mu).dot(A)
    return X
```

[30]:
```python
# (b)
np.random.seed(123)
X = getData(10)
X
```

[30]:
```
array([[-4.44340291, -1.19207249,  1.26575775, -4.78409871, -3.46656417,
        -1.21726601, -0.01481908,  1.94465955,  1.27404405, -2.0943676 ],
       [-2.97243642,  5.61320046,  9.814494  , -4.07449213, -0.51172859,
         6.21690451,  4.5436826 , 10.46040647, 10.81402437,  1.94452439]])
```

[ ]:
```python
# (c)
np.random.seed(123)
X = getData(10000)
mu_hat = np.mean(X, axis=1)
sigma_hat = 1/(10000-1) * (
    (X - mu_hat.reshape((2, 1))) @
    (X - mu_hat.reshape((2, 1))).T
)
print("mu_hat:", np.round(mu_hat, 4))
print("sigma_hat:\n", np.round(sigma_hat, 4))
```

```
mu_hat: [-1.9695  3.0706]
sigma_hat:
 [[ 5.0391 11.0854]
 [11.0854 25.1818]]
```

[81]:
```
# (d)
import torch
np.random.seed(123)
n = 1000
X = getData(n)
X = torch.tensor(X).T # dim: n x 2

X = X.unsqueeze(axis=-1) # dim: n x 2 x 1
X = X.permute(0, 2, 1) # dim: n x 1 x 2
X = X.repeat(1, n, 1) # dim: n x n x 2

U = X.clone().permute(1, 0, 2) # dim: n x n x 2
K = np.exp(-torch.norm(X - U, dim=-1)**2) # dim: n x n
K = K.numpy()
eigenvalues = np.linalg.eigh(K).eigenvalues
eigenvalues.sort()
eigenvalues = np.round(eigenvalues[::-1][:3], 4)
eigenvalues
```

[81]:
```
array([105.5731,  91.1879,  78.3936])
```