Advanced Data Analysis
Homework Week - 11

Aswin Vijay

1. Question

We are asked to prove,

$$\sum_{i,i\prime=1}^{n} W_{i,i\prime}\|Tx_i - Tx_{i\prime}\|^2 = 2 \text{ tr } (TXLX^TT^T) \tag{1}$$

given,

$$\begin{aligned} X &= (x_1, ..., x_n) \\ L &= D - W \\ D &= \text{diag } \left(\sum_{i,i\prime=1}^{n} W_{i,i\prime} \right) \end{aligned} \tag{2}$$

also we use the properties,

$$\begin{aligned} a^Tb &= \text{tr } (ba^T) \\ TT^T &= I_m \\ W_{i,i\prime} &= W_{i\prime,i} \end{aligned} \tag{3}$$

Starting with LHS of (1),

$$\begin{aligned} &\sum_{i,i\prime=1}^{n} W_{i,i\prime}\|Tx_i - Tx_{i\prime}\|^2 \\ &= \sum_{i,i\prime=1}^{n} W_{i,i\prime}(Tx_i - Tx_{i\prime})^T \cdot (Tx_i - Tx_{i\prime}) \\ &= \sum_{i,i\prime=1}^{n} W_{i,i\prime}(x_i^TT^TTx_i - x_i^TT^TTx_{i\prime} - x_{i\prime}^TT^TTx_i + x_{i\prime}^TT^TTx_{i\prime}) \\ &= \sum_{i=1}^{n}\sum_{i\prime=1}^{n} W_{i,i\prime}(\text{tr } (Tx_ix_i^TT^T) + \text{tr } (Tx_{i\prime}x_{i\prime}^TT^T) - \text{tr } (Tx_ix_{i\prime}^TT^T) - \text{tr } (Tx_{i\prime}x_i^TT^T)) \\ &= 2\sum_{i=1}^{n}\sum_{i=1}^{n} W_{i,i}(\text{tr } (Tx_ix_i^TT^T)) - 2\sum_{i=1}^{n}\sum_{i\prime=1}^{n} W_{i,i\prime} \text{ tr } (Tx_ix_{i\prime}^TT^T), \text{using property of trace, (3)} \\ &= 2(\text{tr } (TXDX^TT^T) - \text{tr } (TXWX^TT^T)) \text{ using (2)} \\ &= 2 \text{ tr } (TXLX^TT^T) \end{aligned} \tag{4}$$

Homework Week 11

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import eigh
np.random.seed(1)
```

In [ ]:
```
def data_generation1(n=100):
    return np.concatenate([np.random.randn(n, 1) * 2, np.random.randn(n, 1)], axis=

def data_generation2(n=100):
    return np.concatenate([np.random.randn(n, 1) * 2, 2 * np.round(np.random.rand(
```

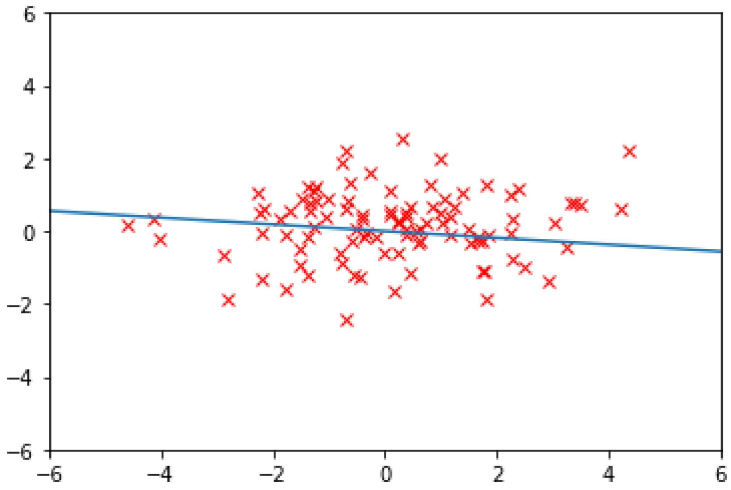
In [ ]:
```
def pca(x, n_components=1):
    x = x - np.mean(x, axis=0)
    w, v = np.linalg.eig(x.T.dot(x))

    return w[:n_components], v[:n_components, :]
```

In [ ]:
```
def llp(x, n_components):
    x = x - np.mean(x, axis=0)
    W = np.exp(-np.sum((x[:, None] - x[None]) ** 2, axis=2))

    D = np.diag(np.sum(W, axis=1))
    L = D - W
    z = x.T @ D @ x
    z = (z + z.T) / 2

    w, v = eigh(x.T @ L @ x, z, eigvals_only=False)

    return w[:n_components], v[:n_components, :]
```

Data 1

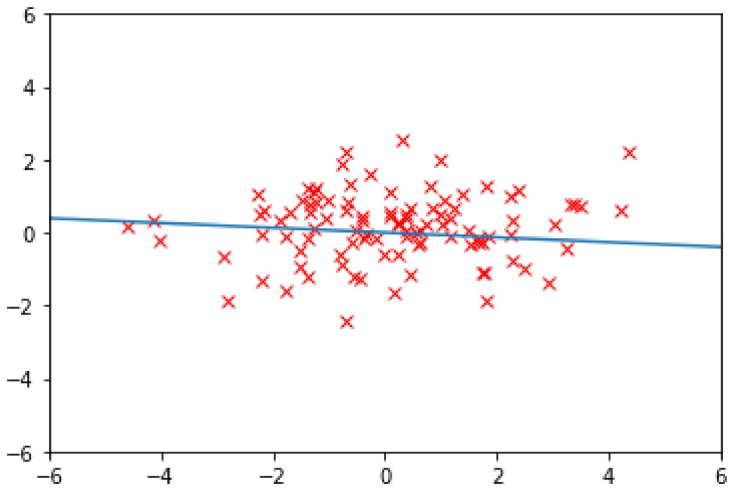
LLP

In [ ]:
```
n = 100
n_components = 1
x = data_generation1(n)
#x = data_generation2(n)
w, v = llp(x, n_components)
#w, v = pca(x, n_components)
plt.xlim(-6., 6.); plt.ylim(-6., 6.)
plt.plot(x[:, 0], x[:, 1], 'rx')
plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
plt.show()
#plt.savefig('example.png')
```



PCA

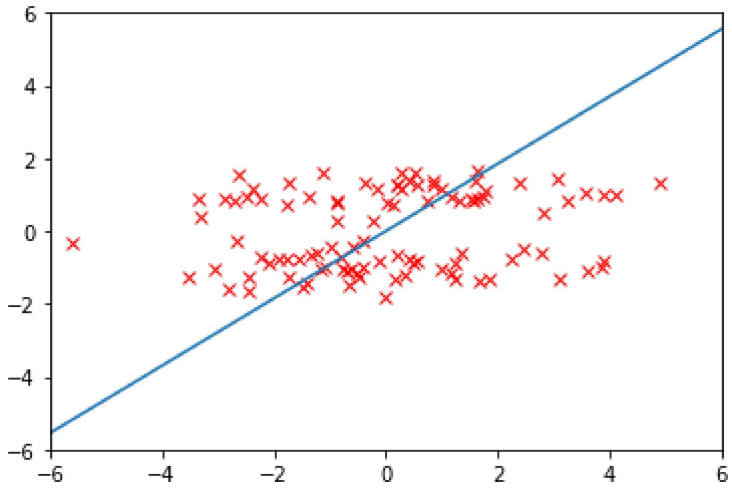
In [ ]:
```
w, v = pca(x, n_components)
plt.xlim(-6., 6.); plt.ylim(-6., 6.)
plt.plot(x[:, 0], x[:, 1], 'rx')
plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
plt.show()
#plt.savefig('example.png')
```



Data 2

LLP

In [ ]:
```
x = data_generation2(n)
w, v = llp(x, n_components)
#w, v = pca(x, n_components)
plt.xlim(-6., 6.); plt.ylim(-6., 6.)
plt.plot(x[:, 0], x[:, 1], 'rx')
plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
plt.show()
#plt.savefig('example.png')
```



PCA

In [ ]:
```
w, v = pca(x, n_components)
plt.xlim(-6., 6.); plt.ylim(-6., 6.)
plt.plot(x[:, 0], x[:, 1], 'rx')
plt.plot(np.array([-v[:, 0], v[:, 0]]) * 900, np.array([-v[:, 1], v[:, 1]]) * 900)
plt.show()
#plt.savefig('example.png')
```