HW02 – Q1

Yue Xu

(a) ∵ X and Y are two zero-mean jointy

∴ $E(X)=0$, $E(Y)=0$

∵ $k=\begin{bmatrix} 5 & 2 \\ 2 & 4 \end{bmatrix}$ ∴ $Cov(X,Y)=Cov(Y,X)=2$

$Var(X)=5$, $Var(Y)=4$

∵ $Cov(X,Y)=E(XY)-E(X)E(Y)=2 \Rightarrow E(XY)=2$

$Var(X)=E(X^2)-E^2(X)=5 \Rightarrow E(X^2)=5$

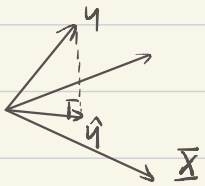
$Var(Y)=E(Y^2)-E^2(Y)=4 \Rightarrow E(Y^2)=4$

∵ when minimize the MSE, $Y-\hat{Y}$ and X are orthoganial

∴ $E[(Y-\hat{Y})X]=0$

∴ $E[(Y-\hat{Y})(X-\underline{E(X)}^{=0})]=0$

∴ $E[XY-Y\cdot E(X)-X\hat{Y}+\hat{Y}E(X)]$



$=E(XY)-\underline{E(Y)}\cdot\underline{E(X)}^{=0}-E(w^*X^2)+E(w^*X)E(X)$

$=E(XY)-w^*E(X^2)+w^*\underline{E^2(X)}^{=0}$

$=E(XY)-w^*E(X^2)$

$=0$

∴ $E(XY)=w^*E(X^2)$

∴ $w^*=\frac{E(XY)}{E(X^2)}=\frac{2}{5}$

(b) $E[(Y-\hat{Y})^2] = E[(Y-\hat{Y})(Y-w^*X)]$

$= E[(Y-\hat{Y})Y - w^*X(Y-\hat{Y})]$

($X(Y-\hat{Y})$: orthogonial)

$= E[Y^2 - \hat{Y}Y]$

$= E(Y^2) - E(w^*XY)$

$= E(Y^2) - w^*E(XY)$

$= 4 - \frac{2}{5} \times 2$

$= \frac{16}{5}$

(c) Suppose $Z = Y - cX \Rightarrow Y = Z + cX$

$\therefore Cov(Z, X) = Cov(Y-cX, X) = R_{XY} - cR_X = 0$

$\therefore c = \frac{R_{XY}}{R_X}$

$\therefore E[Y|X] = E[Z+cX|X]$

$= E[Z|X] + cX$

$= \underline{E(Y)}^{=0} + c(X - \underline{E(X)}^{=0})$

$= \frac{R_{XY}}{R_X} X$

$= \frac{2}{5} = w^*X$

(d) ① ∵ R_X, R_{XY} are second derivative without relation with jointly Gaussian

∴ $\hat{w} = w^* = \frac{2}{5}$

② ∵ X, Y are not jointly Gaussian

∴ $E[Y|X]$ might be nonlinear

∴ MMSE is different

HW02 - Q2

(a) $\because Ke = \lambda e$

$\therefore (K - \lambda I)e = 0$

$$\therefore \begin{bmatrix} 4-\lambda & -1 & 2 \\ -1 & 5-\lambda & -1 \\ 2 & -1 & 3-\lambda \end{bmatrix} \Rightarrow$$

$$(4-\lambda)[(5-\lambda)(3-\lambda)-1] + [-(3-\lambda)+2] + 2[1-2(5-\lambda)]$$

$$= (4-\lambda)(14 - 8\lambda + \lambda^2) + (\lambda - 1) + 2(2\lambda - 9)$$

$$= 56 - 32\lambda + 4\lambda^2 - 14\lambda + 8\lambda^2 - \lambda^3 + 5\lambda - 19$$

$$= -\lambda^3 + 12\lambda^2 - 41\lambda + 37$$

$$= 0$$

$\therefore \lambda_1 = 1.4288$, $\lambda_2 = 3.8567$, $\lambda_3 = 6.7144$

① when $\lambda_1 = 1.4288$, $e_1^T = (0.6346, -0.0574, 0.7708)$

② when $\lambda_2 = 3.8567$, $e_2^T = (-0.4918, -0.7984, 0.3471)$

③ when $\lambda_3 = 6.7144$, $e_3^T = (-0.5962, 0.5994, 0.5341)$

(b) $K = \sum_{k=1}^{3} \lambda_k e_k e_k^T$

$$= 1.4288 \begin{bmatrix} 0.6346 \\ -0.0574 \\ 0.7708 \end{bmatrix} [0.6346, -0.0574, 0.7708]$$

$$+ 3.8567 \begin{bmatrix} -0.4918 \\ -0.7984 \\ 0.3471 \end{bmatrix} [-0.4918, -0.7984, 0.3471]$$

$$+ 6.7144 \begin{bmatrix} -0.5962 \\ 0.5994 \\ 0.5341 \end{bmatrix} [-0.5962, 0.5994, 0.5341]$$

$$= \begin{bmatrix} 4 & -1 & 2 \\ -1 & 5 & -1 \\ 2 & -1 & 3 \end{bmatrix}$$

(c) $X = z_1 e_1 + z_2 e_2 + z_3 e_3$

$$= z_1 \begin{bmatrix} 0.6346 \\ -0.0574 \\ 0.7708 \end{bmatrix} + z_2 \begin{bmatrix} -0.4918 \\ -0.7984 \\ 0.3471 \end{bmatrix} + z_3 \begin{bmatrix} -0.5962 \\ 0.5984 \\ 0.5341 \end{bmatrix}$$

(d) $\tilde{X} = z_2 e_2 + z_3 e_3$

$$= z_2 \begin{bmatrix} -0.4918 \\ -0.7984 \\ 0.3471 \end{bmatrix} + z_3 \begin{bmatrix} -0.5962 \\ 0.5984 \\ 0.5341 \end{bmatrix}$$

(e) $X = z_1 e_1 + z_2 e_2 + z_3 e_3$

$\tilde{X} = z_2 e_2 + z_3 e_3$

$\therefore E[\|X - \tilde{X}\|^2] = E[\|z_1 e_1\|^2] = \lambda_1 = 1.4288$

Q4 Raman spectrum

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
from scipy.interpolate import splev, splrep
```

In [4]:
```
#read raw data
raman=pd.read_table('raman.txt',header=None)
raman_array=raman.to_numpy()
dis=60  #hei=0
peaks,_=find_peaks(raman_array[:,1],distance=dis)
raman_peak_list=[]
for i in peaks:
    raman_peak_list.append(raman_array[i])
raman_peak_array=np.array(raman_peak_list)
#print(raman_peak_array)


#spline
spl=splrep(raman_peak_array[:,0],raman_peak_array[:,1])
#spl=splrep(raman_peak_array[:,0],raman_peak_array[:,1])
x=np.linspace(500,3500,30000)
y=splev(x,spl)
dy=splev(x,spl,der=1)


#find maximum
raman_max_wavenumber=[]
raman_max_intensity=[]
#threshold_max=1.5  #and abs(dy[i])<threshold_max
threshold_min=0.07
prior_i=0
for i in range(1,len(dy)-1):
    if abs(dy[i])>threshold_min and dy[i-1]>0 and dy[i+1]<0:
        #print(i)
        #Take the smaller derivative of two adjacent points
        if i==prior_i+1:
            #print("next")
            if abs(dy[i])>abs(dy[prior_i]):
                continue
            else:
                raman_max_wavenumber.pop()
                raman_max_intensity.pop()
                raman_max_wavenumber.append(x[i])
                raman_max_intensity.append(y[i])
                continue
        raman_max_wavenumber.append(x[i])
        raman_max_intensity.append(y[i])
        prior_i=i
```

(a) Print the wavenumber estimates for the eight largest spectral peak to STDOUT sorted by magnitude.

In [34]:

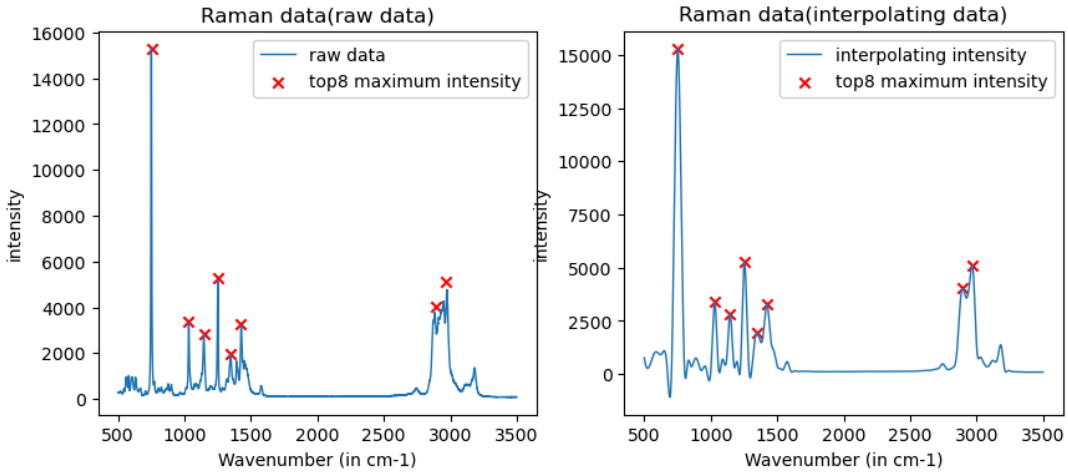
```
raman_peak_sortU=np.argsort(raman_max_intensity)
raman_peak_sortD=raman_peak_sortU[::-1]
raman_top8max_wavenumber=[]
raman_top8max_intensity=[]
print("The eight largest spectral peak:")
for i in range(8):
    raman_top8max_wavenumber.append(raman_max_wavenumber[raman_peak_sortD[i]])
    raman_top8max_intensity.append(raman_max_intensity[raman_peak_sortD[i]])
    print("No",i+1,": wavenumber:",round(raman_top8max_wavenumber[i],5),' inte
```

```
The eight largest spectral peak:
No 1 : wavenumber: 751.70839  intensity: 15300.54001
No 2 : wavenumber: 1253.52512  intensity: 5279.87529
No 3 : wavenumber: 2967.28224  intensity: 5096.37
No 4 : wavenumber: 2895.57985  intensity: 4024.9314
No 5 : wavenumber: 1030.31768  intensity: 3374.3658
No 6 : wavenumber: 1423.73079  intensity: 3265.08494
No 7 : wavenumber: 1145.72152  intensity: 2841.61916
No 8 : wavenumber: 1347.92826  intensity: 1946.51166
```

(b) Create a figure that shows the Raman data and mark each of the maximum intensity values

In [21]:

```
plt.figure(figsize=(10,4))
#raw data
plt.subplot(1, 2, 1)
plt.plot(raman_array[:,0],raman_array[:,1],linewidth=1)
plt.scatter(raman_top8max_wavenumber,raman_top8max_intensity,marker='x',color=
plt.xlabel("Wavenumber (in cm-1)")
plt.ylabel("intensity")
plt.title("Raman data(raw data)")
plt.legend(['raw data','top8 maximum intensity'])

#interpolating data
plt.subplot(1, 2, 2)
plt.plot(x,y,linewidth=1)
plt.scatter(raman_top8max_wavenumber,raman_top8max_intensity,marker='x',color=
plt.xlabel("Wavenumber (in cm-1)")
plt.ylabel("intensity")
plt.title("Raman data(interpolating data)")
plt.legend(['interpolating intensity','top8 maximum intensity'])

plt.show()
```

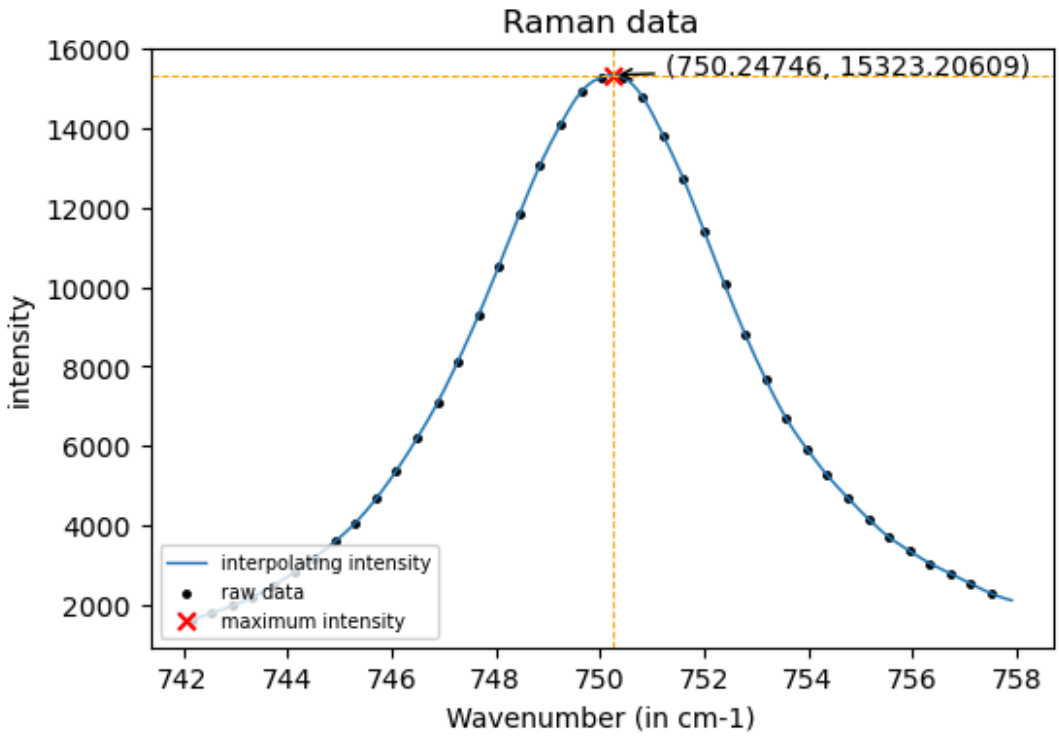


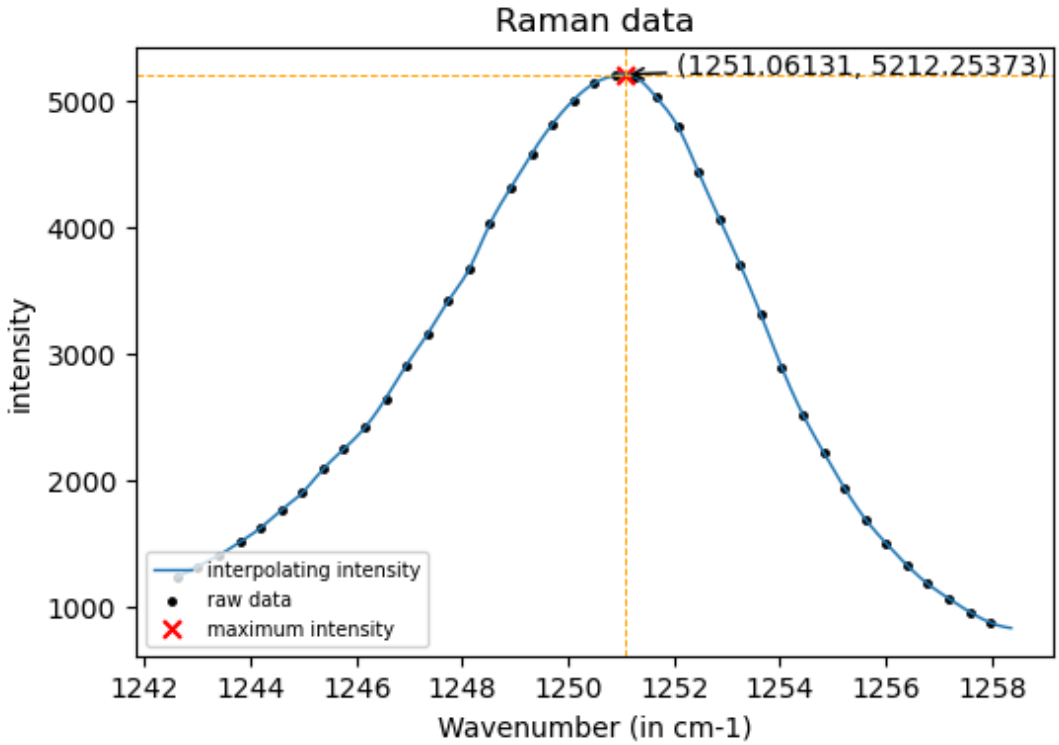
(c) Produce a “zoomed-in” figure for the “regions of interest” corresponding to the four largest peaks. Plot the raw spectral data and overlay your interpolating function. Use a marker to show the wavenumber with maximal intensity

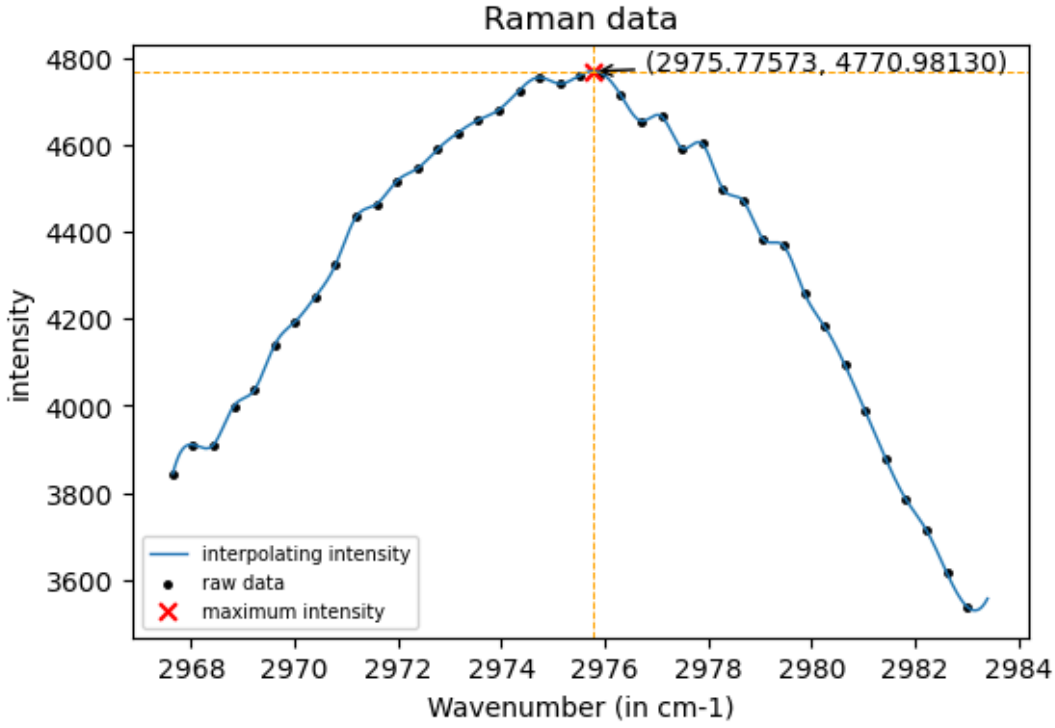
In [45]:

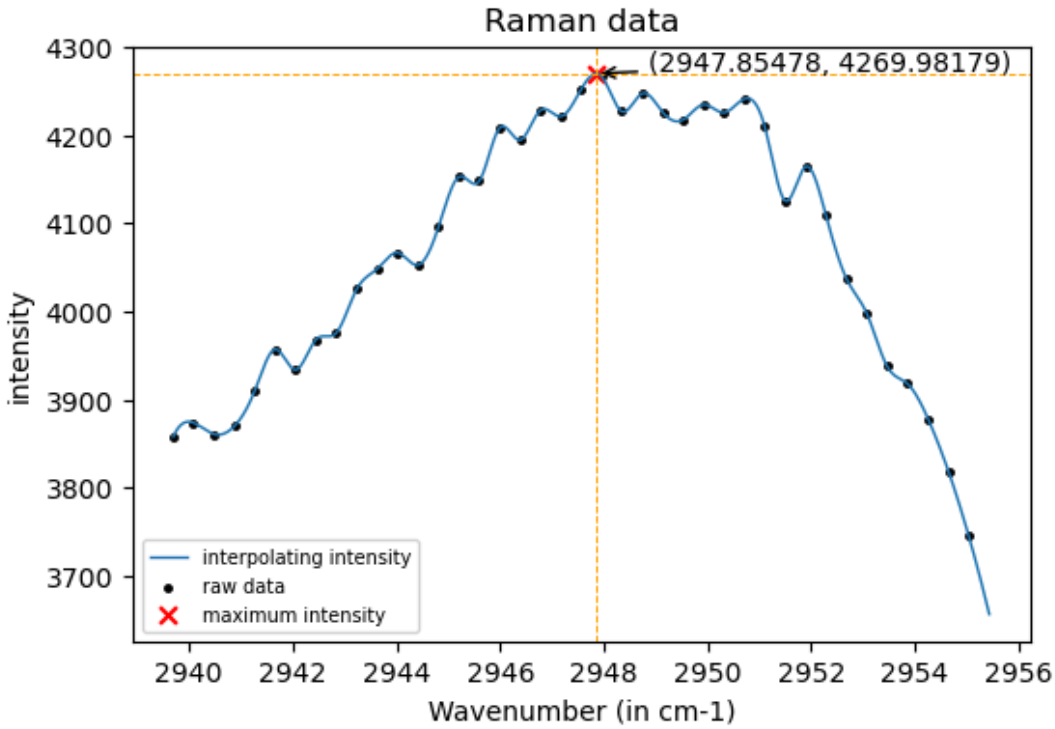
```
#detect top peaks
dis=60
hei=2000
peaks,_=find_peaks(raman_array[:,1],distance=dis,height=hei)
raman_peak_list=[]
raman_peak_index=[]
for i in peaks:
    raman_peak_list.append(raman_array[i])
    raman_peak_index.append(i)
raman_peak_array=np.array(raman_peak_list)
#print(raman_peak_array)

peak_sortU=np.argsort(raman_peak_array[:,1])
peak_sortD=peak_sortU[::-1]

#interpolate intensity within each region of interest
for i in range(4):
    w=20
    l_range=raman_peak_index[peak_sortD[i]]-1-w
    r_range=raman_peak_index[peak_sortD[i]]-1+w
    #print(raman_peak_index[peak_sortD[i]],l_range,r_range)
    #spline
    spl=splrep(raman_array[l_range:r_range,0],raman_array[l_range:r_range,1])
    x=np.linspace(raman_array[l_range,0],raman_array[r_range,0],200*w)
    y=splev(x,spl)
    dy=splev(x,spl,der=1)

    #find maximum
    max_wavenumber=[]
    max_intensity=[]
    #threshold_max=1.5  #and abs(dy[i])<threshold_max
    threshold_min=0.07
    prior_i=0
    for i in range(1,len(dy)-1):
        if abs(dy[i])>threshold_min and dy[i-1]>0 and dy[i+1]<0:
            #print(i)
            #Take the smaller derivative of two adjacent points
            if i==prior_i+1:
                #print("next")
                if abs(dy[i])>abs(dy[prior_i]):
                    continue
                else:
                    max_wavenumber.pop()
                    max_intensity.pop()
                    max_wavenumber.append(x[i])
                    max_intensity.append(y[i])
                    continue
            max_wavenumber.append(x[i])
            max_intensity.append(y[i])
            prior_i=i

    top1max_index=np.argmax(max_intensity)
    top1max_x=max_wavenumber[top1max_index]
    top1max_y=max_intensity[top1max_index]

    plt.figure(figsize=(6,4))
    plt.plot(x,y,linewidth=1)
    plt.scatter(raman_array[l_range:r_range,0],raman_array[l_range:r_range,1],
    plt.scatter(top1max_x,top1max_y,marker='x',color='red',s=40)
    plt.axhline(y=top1max_y,color='orange',linestyle='--',linewidth=0.7)
    plt.axvline(x=top1max_x,color='orange',linestyle='--',linewidth=0.7)
```

```
    plt.annotate(f'({top1max_x:.5f}, {top1max_y:.5f})',
                 xy=(top1max_x, top1max_y),
                 xytext=(top1max_x + 1, top1max_y + 2),
                 arrowprops=dict(facecolor='gray', arrowstyle='->'))
    plt.xlabel("Wavenumber (in cm-1)")
    plt.ylabel("intensity")
    plt.title("Raman data")
    plt.legend(['interpolating intensity', 'raw data','maximum intensity'],fon
    plt.show()
```






Raman data
(2975.77573, 4770.98130)
intensity
Wavenumber (in cm-1)
interpolating intensity
raw data
maximum intensity


Raman data
(2947.85478, 4269.98179)
intensity
Wavenumber (in cm-1)
interpolating intensity
raw data
maximum intensity