1 Proof of question 4

The zero-mean constraint forces $\hat{\boldsymbol{X}}$ to be

$$\hat{\boldsymbol{X}}^T = (\boldsymbol{X}^T, \boldsymbol{x}^T, -\boldsymbol{x}^T)$$

and there's also a constraint on $\boldsymbol{x}$, be aware that $\boldsymbol{x}$ is a row vector with shape $1 \times N$

$$|| \boldsymbol{x} ||_2 = \boldsymbol{x}\boldsymbol{x}^T = 1$$

We conclude that

$$\begin{aligned} \Gamma_{\hat{\boldsymbol{X}}} &= \frac{1}{N+1} \hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} \\ &= \frac{1}{N+1} (\boldsymbol{X}^T \boldsymbol{X} + 2\boldsymbol{x}^T \boldsymbol{x}) \\ &= \frac{1}{N+1} [(N-1)\Gamma_{\boldsymbol{X}} + 2\boldsymbol{x}^T \boldsymbol{x}] \end{aligned}$$

So

$$\begin{aligned} \underset{|| \boldsymbol{x} ||_2 = 1}{\arg\max} \left\| \Gamma_{\hat{\boldsymbol{X}}} - \Gamma_{\boldsymbol{X}} \right\|_F^2 &= \underset{|| \boldsymbol{x} ||_2 = 1}{\arg\max} \left\| \Gamma_{\hat{\boldsymbol{X}}} - \Gamma_{\boldsymbol{X}} \right\|_F^2 \\ &= \underset{|| \boldsymbol{x} ||_2 = 1}{\arg\max} \left\| \boldsymbol{x}^T \boldsymbol{x} - \Gamma_{\boldsymbol{X}} \right\|_F^2 \end{aligned}$$

Notice the equation of function tr

$$\mathrm{tr}(\boldsymbol{X}^T \boldsymbol{X}) = \mathrm{tr}(\boldsymbol{X}\boldsymbol{X}^T) = \|\boldsymbol{X}\|_F^2$$

and

$$\mathrm{tr}(\boldsymbol{A} + \boldsymbol{B}) = \mathrm{tr}(\boldsymbol{A}) + \mathrm{tr}(\boldsymbol{B})$$

tr is exchangeable for matrix multiplication between two matrixes

$$\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{B}\boldsymbol{A})$$

moreover

$$\operatorname{tr}(\boldsymbol{ABC}) = \operatorname{tr}(\boldsymbol{CAB}) = \operatorname{tr}(\boldsymbol{BCA})$$

In fact, due to the symmetry of $\Gamma_{\boldsymbol{X}}$

$$\begin{aligned}
&\underset{\|\boldsymbol{x}\|_2=1}{\arg\max}\left\|\boldsymbol{x}^T\boldsymbol{x} - \Gamma_{\boldsymbol{X}}\right\|_F^2 \\
&\quad= \underset{\|\boldsymbol{x}\|_2=1}{\arg\max}\{\operatorname{tr}((\boldsymbol{x}^T\boldsymbol{x} - \Gamma_{\boldsymbol{X}})^T(\boldsymbol{x}^T\boldsymbol{x} - \Gamma_{\boldsymbol{X}}))\} \\
&\quad= \underset{\|\boldsymbol{x}\|_2=1}{\arg\max}\{\operatorname{tr}(\boldsymbol{x}^T\boldsymbol{x}) - \operatorname{tr}(\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T\boldsymbol{x}) - \operatorname{tr}(\boldsymbol{x}^T\boldsymbol{x}\Gamma_{\boldsymbol{X}}) + \operatorname{tr}(\Gamma_{\boldsymbol{X}}^2)\} \\
&\quad= \underset{\|\boldsymbol{x}\|_2=1}{\arg\max}\{\operatorname{tr}(\boldsymbol{x}^T\boldsymbol{x}) - \operatorname{tr}(\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T\boldsymbol{x}) - \operatorname{tr}(\boldsymbol{x}^T\boldsymbol{x}\Gamma_{\boldsymbol{X}})\} \\
&\quad= \underset{\|\boldsymbol{x}\|_2=1}{\arg\max}\{\operatorname{tr}(\boldsymbol{x}\boldsymbol{x}^T) - 2\operatorname{tr}(\boldsymbol{x}\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T)\} \\
&\quad= \underset{\|\boldsymbol{x}\|_2=1}{\arg\max}\{1 - 2\boldsymbol{x}\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T\} \\
&\quad= \underset{\|\boldsymbol{x}\|_2=1}{\arg\min}\{\boldsymbol{x}\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T\}
\end{aligned}$$

Construct Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \lambda) := \boldsymbol{x}\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T + \lambda(1 - \boldsymbol{x}\boldsymbol{x}^T)$$

and take derivative to the formulation

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{x}^T} = 2\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T - 2\lambda\boldsymbol{x}^T$$

and

$$\frac{\partial\mathcal{L}}{\partial\lambda} = \boldsymbol{x}\boldsymbol{x}^T - 1$$

The optimal solution $\boldsymbol{x}^T$ satisfies

$$\Gamma_{\boldsymbol{X}}\boldsymbol{x}^T = \lambda\boldsymbol{x}^T$$

$\boldsymbol{x}^T$ is the eigenvector of the matrix $\Gamma_{\boldsymbol{X}}$ and λ is the related eigenvalue. When reaching the optimal value, the expression of $\mathcal{L}$ must be

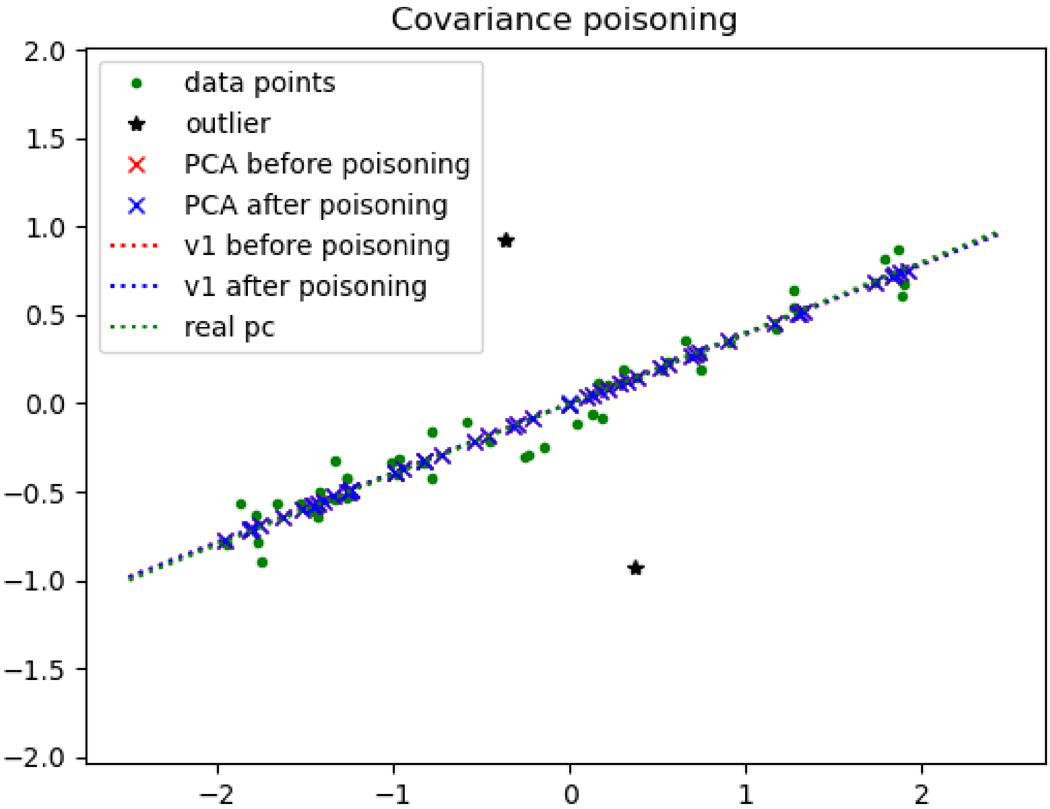
$$\mathcal{L}(\boldsymbol{x}, \lambda) := \boldsymbol{x}\lambda\boldsymbol{x}^T = \lambda\boldsymbol{x}\boldsymbol{x}^T = \lambda$$

So the optimal solution $\boldsymbol{x}^T$ is identical to the unit eigenvector of $\Gamma_{\boldsymbol{X}}$ correspond to the minimum eigenvalue.

Follow the proof, it's easy to design the python program in question 4

```
import numpy as np
"""
numpy.linalg.eig(a)

    Compute the eigenvalues and right eigenvectors of a square array.

Parameters:
    a: (..., M, M) array
    Matrices for which the eigenvalues and right eigenvectors will be computed.

Returns:
    w:  (..., M) array
    The eigenvalues are not necessarily ordered. The resulting array will be of complex type, unless the imaginary part is zero in which case it will be cast to a real type. When a is real the resulting eigenvalues will be real (0 imaginary part) or occur in conjugate pairs.

    v:  (..., M, M) array
    The normalized (unit “length”) eigenvectors, such that the column v[:,i] is the eigenvector corresponding to the eigenvalue w[i].
"""
def coupled_outlier_manipulation(xs: np.ndarray):
    N, D = xs.shape
    cov = xs.T @ xs / (N - 1)
    eig, V = np.linalg.eig(cov)
    x = V[:,np.argmin(eig)]
    x = x / np.linalg.norm(x, ord=2)
    outliers = np.stack((x, -x))
    return outliers, np.concatenate((xs, outliers))
```

and the outliers are easy to discover in the visualization of the result


Covariance poisoning
data points
outlier
PCA before poisoning
PCA after poisoning
v1 before poisoning
v1 after poisoning
real pc
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

□