# Exercise Sheet 6 - Solutions for Theory Part

Bolkar EREN
Muhammet Ali AL
Maleyka JABBARLI
Aycan CELIK

### Exercise 1

We applied the Lagrange Multipliers to the given optimization problem in the lecture06b, so let us remember what we have covered, the definitions and the Lagrangian:

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}) := \boldsymbol{w}_x^\top C_{xy} \boldsymbol{w}_{\boldsymbol{y}} + \frac{1}{2}\boldsymbol{\lambda}_x \cdot (1 - \boldsymbol{w}_x^\top C_{xx} \boldsymbol{w}_x) + \frac{1}{2}\boldsymbol{\lambda}_y \cdot (1 - \boldsymbol{w}_y^\top C_{yy} \boldsymbol{w}_y) \\ J(\boldsymbol{w}) := \boldsymbol{w}_x^\top C_{xy} \boldsymbol{w}_{\boldsymbol{y}} \\ g_1(\boldsymbol{w}) := 1 - \boldsymbol{w}_x^\top C_{xx} \boldsymbol{w}_x \\ g_2(\boldsymbol{w}) := 1 - \boldsymbol{w}_y^\top C_{yy} \boldsymbol{w}_y\end{aligned}$$

This converts the optimization problem into the Lagrangian Multipliers Method which maximizes $J(\boldsymbol{w}) = \boldsymbol{w}_x^\top C_{xy} \boldsymbol{w}_{\boldsymbol{y}}$ restricted on the multiple constraints of $0 = g_1(\boldsymbol{w}) = 1 - \boldsymbol{w}_x^\top C_{xx} \boldsymbol{w}_x$ and $0 = g_2(\boldsymbol{w}) = 1 - \boldsymbol{w}_y^\top C_{yy} \boldsymbol{w}_y$.

Setting the gradient to zero

$$\nabla_{\boldsymbol{w}} \mathcal{L} = 0 \implies \nabla_{\boldsymbol{w}_x} \mathcal{L} = 0 \quad \& \quad \nabla_{\boldsymbol{w}_y} \mathcal{L} = 0$$

and using the properties from The Matrix Cookbook and:

$$\begin{aligned}0 &= \nabla_{\boldsymbol{w}_x} \mathcal{L} \\ &= \nabla_{\boldsymbol{w}_x} (\boldsymbol{w}_x^\top C_{xy} \boldsymbol{w}_{\boldsymbol{y}}) + \nabla_{\boldsymbol{w}_x} (\frac{1}{2}\boldsymbol{\lambda}_x \cdot (1 - \boldsymbol{w}_x^\top C_{xx} \boldsymbol{w}_x)) \\ &\quad + \nabla_{\boldsymbol{w}_x} (\frac{1}{2}\boldsymbol{\lambda}_y \cdot (1 - \boldsymbol{w}_y^\top C_{yy} \boldsymbol{w}_y)) \\ &= C_{xy} \boldsymbol{w}_y + \frac{1}{2}\boldsymbol{\lambda}_x \cdot (-2 C_{xx} \boldsymbol{w}_x) + 0 \\ &= C_{xy} \boldsymbol{w}_y - \boldsymbol{\lambda}_x C_{xx} \boldsymbol{w}_x\end{aligned}$$

which implies

$$C_{xy} \boldsymbol{w}_y = \boldsymbol{\lambda}_x C_{xx} \boldsymbol{w}_x \tag{1}$$

2and we also have the other part as:

$$
\begin{aligned}
0 &= \nabla_{\boldsymbol{w}_y}\mathcal{L} \\
&= \nabla_{\boldsymbol{w}_y}(\boldsymbol{w}_x^\top C_{xy}\boldsymbol{w_y}) + \nabla_{\boldsymbol{w}_y}(\frac{1}{2}\boldsymbol{\lambda}_x \cdot (1 - \boldsymbol{w}_x^\top C_{xx}\boldsymbol{w}_x)) \\
&\quad + \nabla_{\boldsymbol{w}_y}(\frac{1}{2}\boldsymbol{\lambda}_y \cdot (1 - \boldsymbol{w}_y^\top C_{yy}\boldsymbol{w}_y)) \\
&= \nabla_{\boldsymbol{w}_y}(\boldsymbol{w}_x^\top C_{yx}^\top \boldsymbol{w_y}) + 0 + \frac{1}{2}\boldsymbol{\lambda}_y \cdot (-2C_{yy}\boldsymbol{w}_y) \\
&= \nabla_{\boldsymbol{w}_y}((C_{yx}\boldsymbol{w}_x)^\top \boldsymbol{w_y}) - \boldsymbol{\lambda}_y C_{yy}\boldsymbol{w}_y \\
&= C_{yx}\boldsymbol{w}_x - \boldsymbol{\lambda}_y C_{yy}\boldsymbol{w}_y
\end{aligned}
$$

which implies

$$
C_{yx}\boldsymbol{w}_x = \boldsymbol{\lambda}_y C_{yy}\boldsymbol{w}_y \tag{2}
$$

Thus, a solution of the original problem must necessarily satisfy these two equations (1) and (2). We will use these equations to prove **a)** very soon.

In addition, we applied matrix multiplication to the first equation (1), multiplying both sides by $\boldsymbol{w}_x^\top$ from left, and we also applied matrix multiplication to the second equation (2), multiplying both sides by $\boldsymbol{w}_y^\top$ from left. The dimensions were compatible, so we obtained the following results:

$$
\begin{aligned}
J(\boldsymbol{w}) &= \boldsymbol{w}_x^\top C_{xy}\boldsymbol{w}_y = \boldsymbol{w}_x^\top(\boldsymbol{\lambda}_x C_{xx}\boldsymbol{w}_x) = \boldsymbol{\lambda}_x(\boldsymbol{w}_x^\top C_{xx}\boldsymbol{w}_x) = \boldsymbol{\lambda}_x \cdot 1 = \boldsymbol{\lambda}_x \\
J(\boldsymbol{w}) &= (\boldsymbol{w}_x^\top C_{xy}\boldsymbol{w}_y)^\top \\
&= \boldsymbol{w}_y^\top C_{yx}\boldsymbol{w}_x = \boldsymbol{w}_y^\top(\boldsymbol{\lambda}_y C_{yy}\boldsymbol{w}_y) = \boldsymbol{\lambda}_y(\boldsymbol{w}_y^\top C_{yy}\boldsymbol{w}_y) = \boldsymbol{\lambda}_y \cdot 1 = \boldsymbol{\lambda}_y
\end{aligned}
$$

where the constraints are applied and we used the fact that $J(\boldsymbol{w})$ is a scalar, $1 \times 1$ matrix, so it is invariant under the transpose.

Now we can continue with the exercise sheet.

**a)** Using $\boldsymbol{\lambda} := \boldsymbol{\lambda}_x = \boldsymbol{\lambda}_y = J(\boldsymbol{w})$ and equations (1) and (2) we can write it as a system of equations;

$$
\begin{bmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{bmatrix}\begin{bmatrix} w_x \\ w_y \end{bmatrix} = \begin{bmatrix} 0 \cdot w_x + C_{xy}w_x \\ C_{yx}w_x + 0 \cdot w_y \end{bmatrix} = \begin{bmatrix} \lambda_x C_{xx}w_x \\ \lambda_y C_{yy}w_y \end{bmatrix} = \lambda \begin{bmatrix} C_{xx} & 0 \\ 0 & C_{yy} \end{bmatrix}\begin{bmatrix} w_x \\ w_y \end{bmatrix}
$$

**b)** We observed above that

$$
\lambda_x = J(\boldsymbol{w}) = \lambda_y
$$

for the Lagrange Multipliers $\lambda_x$ and $\lambda_y$, and we defined $\lambda := \lambda_x = \lambda_y = J(\boldsymbol{w})$, then we showed that this $\lambda$ is an eigenvalue where the solution $(\boldsymbol{w}_x, \boldsymbol{w}_y)$ to the original optimization problem is actually an eigenvector to the given generalized eigenvalue problem. We know that the function maximizes $J(\boldsymbol{w})$ and by equality we have $\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \max_{\boldsymbol{w}} \lambda$ so the eigenvector $(w_x, w_y)$ correspond to $\lambda_{\text{max}}$

## Exercise 2

**a)** Suppose that we have the pair $(\boldsymbol{w}'_x, \boldsymbol{w}'_y)$ as an optimal solution for the CCA problem in the question. We will find another optimal solution for this CCA problem where the solution lies in the span of the data.

We have $\text{Im}(X) \subseteq \mathbb{R}^{d_1}$ and $\text{Im}(Y) \subseteq \mathbb{R}^{d_2}$ as the span of the data where we also have the orthogonal complements for the image spaces as $\text{Im}(X)^\top$, $\text{Im}(Y)^\top$ such that $\boldsymbol{w}'_{\boldsymbol{x}} \in \mathbb{R}^{d_1} = \text{Im}(X) + \text{Im}(X)^\top$ and $\boldsymbol{w}'_y \in \mathbb{R}^{d_2} = \text{Im}(Y) + \text{Im}(Y)^\top$. Then we have the sums for both vectors as:

$$\begin{aligned}
\boldsymbol{w}'_{\boldsymbol{x}} &= \boldsymbol{w}_x + \boldsymbol{v}_x, \quad s.t \quad \boldsymbol{w}_x \in \text{Im}(X), \boldsymbol{v}_x \in \text{Im}(X)^\top \\
\boldsymbol{w}'_{\boldsymbol{y}} &= \boldsymbol{w}_y + \boldsymbol{v}_y, \quad s.t. \quad \boldsymbol{w}_y \in \text{Im}(Y), \boldsymbol{v}_y \in \text{Im}(Y)^\top
\end{aligned}$$

and since $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ are in the span of data matrices $X, Y$, then there exist vectors $\boldsymbol{\alpha}_x, \boldsymbol{\alpha}_y \in \mathbb{R}^N$ as:

$$\begin{aligned}
\boldsymbol{w}'_{\boldsymbol{x}} &= X\boldsymbol{\alpha}_x + \boldsymbol{v}_x, \quad s.t \quad \boldsymbol{\alpha}_x \in \mathbb{R}^N, \boldsymbol{v}_x \in \text{Im}(X)^\top \\
\boldsymbol{w}'_{\boldsymbol{y}} &= Y\boldsymbol{\alpha}_y + \boldsymbol{v}_y, \quad s.t \quad \boldsymbol{\alpha}_y \in \mathbb{R}^N, \boldsymbol{v}_y \in \text{Im}(Y)^\top
\end{aligned}$$

Here it is denoted as $\boldsymbol{w}_x := X\boldsymbol{\alpha}_x$ and $\boldsymbol{w}_y := Y\boldsymbol{\alpha}_y$. We will show that these are optimal solutions, too. Now we insert these equations to optimization where we know already that $(w'_x, w'_y)$ is a solution:

$$\begin{aligned}
(w'_x, w'_y) \quad \text{maximizes} \quad & w'_x C_{xy} w'_y \\
s.t \quad & w'_x C_{xx} w'_x = 1 \\
& w'_y C_{yy} w'_y = 1
\end{aligned}$$

We have the following for the objective:

$$\begin{aligned}
w'_x C_{xy} w'_y &= (w_x + v_x) C_{xy} (w_y + v_y) \\
&= w_x C_{xy} w_y + w_x C_{xy} v_y + v_x C_{xy} w_y + v_x C_{xy} v_y \\
&= w_x C_{xy} w_y + 0 + 0 + 0 \\
&= w_x C_{xy} w_y
\end{aligned}$$

where the zeroes comes from the face that $v_x \perp \text{Im}(X)$ and $v_y \perp \text{Im}(Y)$ and $C_{xy} = \frac{1}{N} XY^\top$, so for each term a product with the data matrices is multiplied by $v_x$ or $v_y$ which results 0 expect the first term.

In addition, we have the constraints:

$$\begin{aligned}
1 = w'_x C_{xx} w'_x &= (w_x + v_x) C_{xx} (w_x + v_x) \\
&= w_x C_{xx} w_x + w_x C_{xx} v_x + v_x C_{xx} w_x + v_x C_{xx} v_x \\
&= w_x C_{xx} w_x + 0 + 0 + 0 \\
&= w_x C_{xx} w_x
\end{aligned}$$

and also

$$\begin{aligned}1 = w'_y C_{yy} w'_y &= (w_y + v_y) C_{yy} (w_y + v_y)\\ &= w_y C_{yy} w_y + w_y C_{yy} v_y + v_y C_{yy} w_y + v_y C_{yy} v_y\\ &= w_y C_{yy} w_y + 0 + 0 + 0\\ &= w_y C_{yy} w_y\end{aligned}$$

where we used a similar reasoning as we used for the objective.

However, this leaves us with the following equations:

$$\begin{aligned}(w_x, w_y) \quad \text{satisfies} \quad & w_x C_{xy} w_y = w'_x C_{xy} w'_y\\ s.t \quad & w_x C_{xx} w_x = 1\\ & w_y C_{yy} w_y = 1\end{aligned}$$

where we know that $w'_x C_{xy} w'_y$ is the maximum value for the optimization problem, so the pair $(w_x, w_y)$ actually maximizes the same objective function with the same constraints which means that this new pair is a solution, too.

Furthermore, we have $w_x = X\alpha_x$ and $w_y = Y\alpha_y$ which shows that this new solutions are in the span of the data.

**b)** We start by formulating the Lagrangian with $\boldsymbol{\alpha}, \boldsymbol{\lambda}_x, \boldsymbol{\lambda}_y$.

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) := \boldsymbol{\alpha}_x^\top Q_{xy} \boldsymbol{\alpha}_y + \frac{1}{2}\boldsymbol{\lambda}_x \cdot (1 - \boldsymbol{\alpha}_x^\top Q_{xx} \boldsymbol{\alpha}_x) + \frac{1}{2}\boldsymbol{\lambda}_y \cdot (1 - \boldsymbol{\alpha}_y^\top Q_{yy} \boldsymbol{\alpha}_y)\\ J(\boldsymbol{\alpha}) := \boldsymbol{\alpha}_x^\top Q_{xy} \boldsymbol{\alpha}_y\\ g_1(\boldsymbol{\alpha}) := 1 - \boldsymbol{\alpha}_x^\top Q_{xx} \boldsymbol{\alpha}_x\\ g_2(\boldsymbol{\alpha}) := 1 - \boldsymbol{\alpha}_y^\top Q_{yy} \boldsymbol{\alpha}_y\end{aligned}$$

Setting the gradient of Lagrangian to zero we get following;

$$\begin{aligned}0 &= \nabla_{\boldsymbol{\alpha}_x} \mathcal{L}\\ &= \nabla_{\boldsymbol{\alpha}_x} (\boldsymbol{\alpha}_x^\top Q_{xy} \boldsymbol{\alpha}_y) + \nabla_{\boldsymbol{\alpha}_x} (\frac{1}{2}\boldsymbol{\lambda}_x \cdot (1 - \boldsymbol{\alpha}_x^\top Q_{xx} \boldsymbol{\alpha}_x))\\ &\quad + \nabla_{\boldsymbol{\alpha}_x} (\frac{1}{2}\boldsymbol{\lambda}_y \cdot (1 - \boldsymbol{\alpha}_y^\top Q_{yy} \boldsymbol{\alpha}_y))\\ &= Q_{xy} \boldsymbol{\alpha}_y + \frac{1}{2}\boldsymbol{\lambda}_x \cdot (-2 Q_{xx} \boldsymbol{\alpha}_x) + 0\\ &= Q_{xy} \boldsymbol{\alpha}_y - \boldsymbol{\lambda}_x Q_{xx} \boldsymbol{\alpha}_x\\ &\Rightarrow Q_{xy} \boldsymbol{\alpha}_y = \boldsymbol{\lambda}_x Q_{xx} \boldsymbol{\alpha}_x\end{aligned}$$

$$
\begin{aligned}
0 &= \nabla_{\boldsymbol{\alpha}_y} \mathcal{L} \\
&= \nabla_{\boldsymbol{\alpha}_y} (\boldsymbol{\alpha}_x^\top Q_{xy} \boldsymbol{\alpha}_y) + \nabla_{\boldsymbol{\alpha}_y} (\frac{1}{2} \boldsymbol{\lambda}_x \cdot (1 - \boldsymbol{\alpha}_x^\top Q_{xx} \boldsymbol{\alpha}_x)) \\
&\quad + \nabla_{\boldsymbol{\alpha}_y} (\frac{1}{2} \boldsymbol{\lambda}_y \cdot (1 - \boldsymbol{\alpha}_y^\top Q_{yy} \boldsymbol{\alpha}_y)) \\
&= Q_{yx} \boldsymbol{\alpha}_x + \frac{1}{2} \boldsymbol{\lambda}_y \cdot (-2 Q_{yy} \boldsymbol{\alpha}_y) + 0 \\
&= Q_{yx} \boldsymbol{\alpha}_x - \boldsymbol{\lambda}_y Q_{yy} \boldsymbol{\alpha}_y \\
&\Rightarrow Q_{yx} \boldsymbol{\alpha}_x = \boldsymbol{\lambda}_y Q_{yy} \boldsymbol{\alpha}_y
\end{aligned}
$$

Using the partial derivatives we derive equality between Lagrange multipliers and objective function;

$$
\begin{aligned}
&\Rightarrow Q_{yx} \boldsymbol{\alpha}_x = \boldsymbol{\lambda}_y Q_{yy} \boldsymbol{\alpha}_y \\
&\Rightarrow \boldsymbol{\alpha}_y^\top Q_{yx} \boldsymbol{\alpha}_x = \boldsymbol{\lambda}_y \boldsymbol{\alpha}_y^\top Q_{yy} \boldsymbol{\alpha}_y \\
&\Rightarrow \boldsymbol{\alpha}_y^\top Q_{yx} \boldsymbol{\alpha}_x = \boldsymbol{\lambda}_y \\
&\Rightarrow (\boldsymbol{\alpha}_x^\top Q_{yx} \boldsymbol{\alpha}_y)^\top = \boldsymbol{\lambda}_y
\end{aligned}
$$

$$
\begin{aligned}
&\Rightarrow \boldsymbol{\alpha}_x^\top Q_{yx} \boldsymbol{\alpha}_y = \boldsymbol{\lambda}_y \\
&\Rightarrow Q_{xy} \boldsymbol{\alpha}_y = \boldsymbol{\lambda}_x Q_{xx} \boldsymbol{\alpha}_x \\
&\Rightarrow \boldsymbol{\alpha}_x^\top Q_{xy} \boldsymbol{\alpha}_y = \boldsymbol{\lambda}_x \boldsymbol{\alpha}_x^\top Q_{xx} \boldsymbol{\alpha}_x \\
&\Rightarrow \boldsymbol{\alpha}_x^\top Q_{xy} \boldsymbol{\alpha}_y = \boldsymbol{\lambda}_x
\end{aligned}
$$

Using $\boldsymbol{\lambda} := \boldsymbol{\lambda}_x = \boldsymbol{\lambda}_y = J(\boldsymbol{\alpha})$ we can write the gradient of Lagrangian as a system of equations;

$$
\begin{bmatrix} 0 & Q_{xy} \\ Q_{yx} & 0 \end{bmatrix} \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \boldsymbol{\lambda} \begin{bmatrix} Q_{xx} & 0 \\ 0 & Q_{yy} \end{bmatrix} \begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix}
$$

**c)** Since we showed $J(\boldsymbol{\alpha}) = \boldsymbol{\lambda}$, we have $\max_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} \boldsymbol{\lambda}$ which simply gives us $\lambda_{\text{max}}$ since $\boldsymbol{\alpha}$ are eigenvectors with corresponding eigenvalue $\boldsymbol{\lambda}$.

**d)** Solutions to original problem are defined as linear transformation of solutions to high-dimensional CCA s.t. we have $\boldsymbol{w}_x = X\boldsymbol{\alpha}_{\boldsymbol{x}}$ and $\boldsymbol{w}_y = Y\boldsymbol{\alpha}_{\boldsymbol{y}}$. We only needed to multiply data matrices $X, Y$ with the solution vectors $\alpha_x, \alpha_y$ which are given by the assumption.

Exercises for the course
**Machine Learning for Data Science**
Winter Semester 2024/25

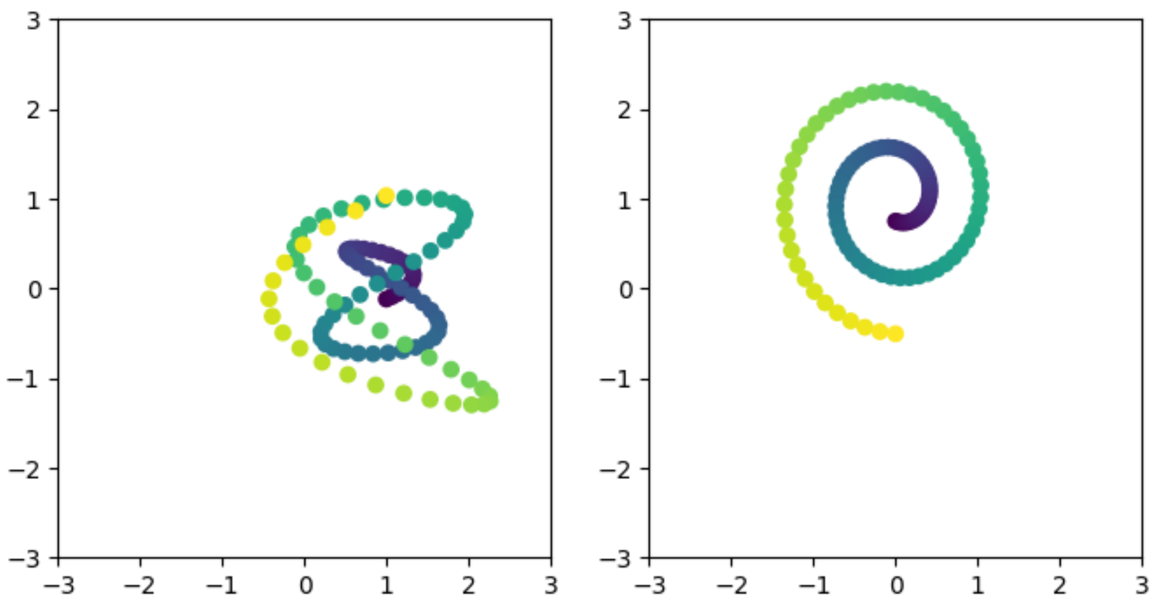
G. Montavon
Institute of Computer Science
**Department of Mathematics and Computer Science**
Freie Universität Berlin

# Exercise Sheet 6 (programming part)

In this exercise, we consider canonical correlation analysis (CCA) on two simple problems, one in low dimensions and one in high dimensions. The goal is to implement the original CCA procedure, and the CCA variant for high-dimensional data, in order to handle both cases. The first dataset consists of two trajectories in two dimensions. The dataset is extracted and plotted below. The first data points are shown in dark blue, and the last ones are shown in yellow.

In [1]:
```
import numpy
import matplotlib
%matplotlib inline
from matplotlib import pyplot as plt
import utils

X,Y   = utils.getdata()
p1,p2 = utils.plotdata(X, Y)
```



For these two trajetories, that can be understood as two different modalities of the same data, we would like determine under which projections they appear maximally correlated.

## Exercise 3: Implementing CCA (25 P)

As stated in the lecture, the CCA problem in its original form consists of maximizing the cross-correlation objective:

$$J(w_x, w_y) = w_x^\top C_{xy} w_y$$

subject to autocorrelation constraints $w_x^\top C_{xx} w_x = 1$ and $w_y^\top C_{yy} w_y = 1$. Using the method of Lagrange multipliers, this optimization problem can be reduced to finding the first eigenvector of the generalized eigenvalue problem:

$$\begin{bmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \end{bmatrix} = \lambda \begin{bmatrix} C_{xx} & 0 \\ 0 & C_{yy} \end{bmatrix} \begin{bmatrix} w_x \\ w_y \end{bmatrix}$$

Your first task is to write a function that solves this generalized eigenvalue problem. The function you need to implement receives two matrices `X` and `Y` of size `d1` $\times$ `N` and `d2` $\times$ `N` respectively. It should return two vectors of size `d1` and `d2` corresponding to the projections associated to the modalities `X` and `Y`. *(Hint: Note that the data matrices `X` and `Y` have not been centered yet.)*

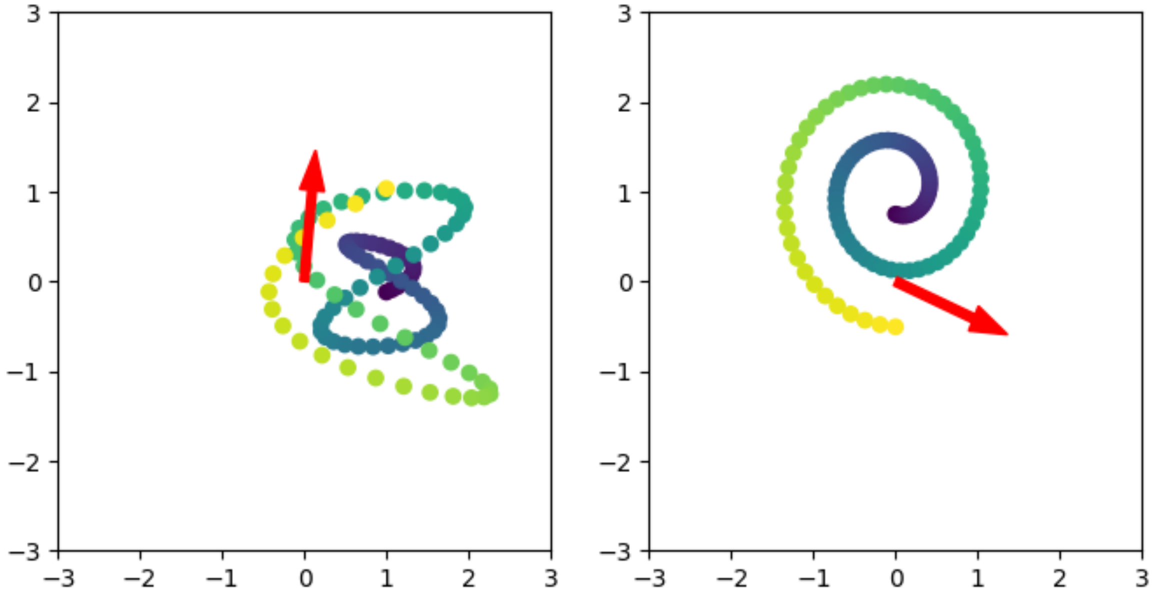
In [2]:
```
import numpy
from scipy.linalg import eigh, eig


def CCA(X,Y):
    ## -------------------------------
    centered_X = (X - numpy.mean(X, axis=1, keepdims=True)) / numpy.std(X
    centered_Y = (Y - numpy.mean(Y, axis=1, keepdims=True)) / numpy.std(Y
    N = X.shape[1]
    d1 = X.shape[0]
    d2 = Y.shape[0]
    A = numpy.zeros(shape=(d1+d2, d1+d2))
    B = numpy.zeros(shape=(d1+d2, d1+d2))
    CXY = centered_X @ centered_Y.T / N
    CXX = centered_X @ centered_X.T / N
    CYY = centered_Y @ centered_Y.T / N
    A[:d1, d1:] = CXY
    A[d2:, :d2] = CXY.T
    B[:d1, :d1] = CXX
    B[d1:, d1:] = CYY
    eigval, eigveg = eigh(A, B)
    lambda_1 = numpy.argmax(eigval)
    wx, wy = eigveg[:d1, lambda_1], eigveg[d1:, lambda_1]
    wx = wx / numpy.sqrt(wx.T @ CXX @ wx)
    wy = wy / numpy.sqrt(wy.T @ CYY @ wy)
    ## -------------------------------
    # import solution
    # wx,wy = solution.CCA(X,Y)
    ## -------------------------------
    return -wx, -wy
```

The function can now be called with our dataset. The learned projection vectors $w_x$ and $w_y$ are plotted as red arrows.

In [3]:
```
wx,wy = CCA(X,Y)
p1,p2 = utils.plotdata(X,Y)
p1.arrow(0,0,1*wx[0],1*wx[1],color='red',width=0.1)
p2.arrow(0,0,1*wy[0],1*wy[1],color='red',width=0.1)
```

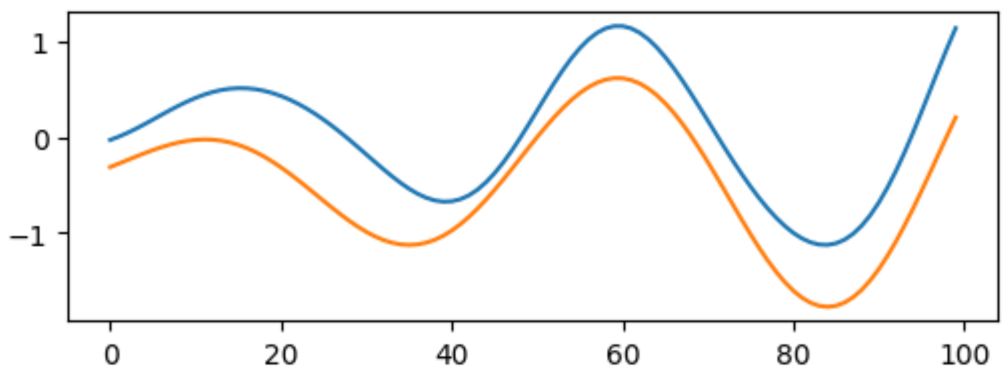
```
plt.show()
```




In each modality, the arrow points in a specific direction (note that the optimal CCA directions are defined up to a sign flip of both $w_x$ and $w_y$). Furthermore, we can verify CCA has learned a meaningful solution by projecting the data on it.

In [4]:
```
plt.figure(figsize=(6,2))

plt.plot(wx.dot(X))
plt.plot(wy.dot(Y))
plt.show()
```



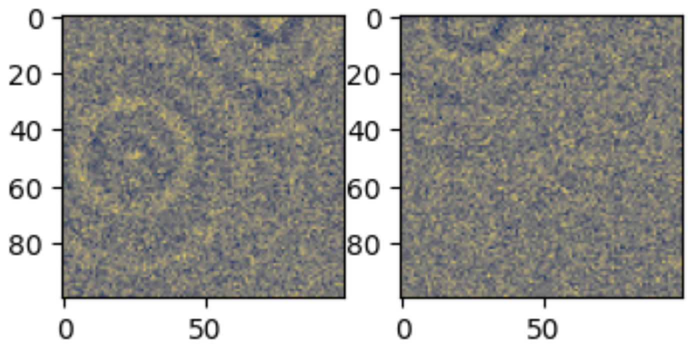

Clearly, the data is correlated in the projected space.

# Exercise 4: Implementing CCA for High Dimensions (25 P)

In the second part of the exercise, we consider the case where the data is high dimensional (with $d \gg N$). Such high-dimensionality occurs for example, when input data are images. We consider the scenario where sources emit spatially, and two (noisy) receivers measure the spatial field at different locations. We would like to identify the signal that is common to the two measured locations, e.g. a given source emitting at a given frequency. We first load the data and show one example.

In [5]:
```
X,Y = utils.getHDdata()

utils.plotHDdata(X[:,0],Y[:,0])
plt.show()
```



Several sources can be perceived, however, there is a significant level of noise. Here again, we will use CCA to find subspaces where the two modalities are maximally correlated. In this example, because there are many more dimensions than there are data points, it is more advantageous to solve the alternate formulation of CCA in terms of the weightings $\alpha_x$ and $\alpha_y$. Your task is to implement the latter CCA solver. Like the original CCA solver, it receives two data matrices of size `d1` $\times$ `N` and `d2` $\times$ `N` respectively as input, and should return the associate CCA directions (two vectors of respective sizes `d1` and `d2` ).

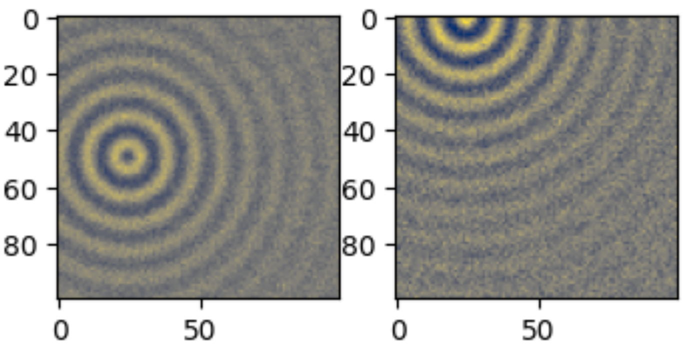
In [6]:
```
def CCA_HD(X,Y):

    ## --------------------------------
    centered_X = (X - numpy.mean(X, axis=1, keepdims=True)) / numpy.std(X
    centered_Y = (Y - numpy.mean(Y, axis=1, keepdims=True)) / numpy.std(Y
    N = X.shape[1]
    A = numpy.zeros(shape=(N+N, N+N))
    B = numpy.zeros(shape=(N+N, N+N))
    XTX = centered_X.T @ centered_X
    YTY = centered_Y.T @ centered_Y
    QXY = (XTX @ YTY) / N
    QXX = (XTX @ XTX) / N
    QYY = (YTY @ YTY) / N
    A[:N, N:] = QXY
    A[N:, :N] = QXY.T
    B[:N, :N] = QXX
    B[N:, N:] = QYY
    B += 0.00001 * numpy.eye(N+N)
    eigval, eigveg = eigh(A, B)
    lambda_1 = numpy.argmax(eigval)
    alpha_x, alpha_y = eigveg[:N, lambda_1], eigveg[N:, lambda_1]
    alpha_x = alpha_x / numpy.sqrt(alpha_x.T @ QXX @ alpha_x)
    alpha_y = alpha_y / numpy.sqrt(alpha_y.T @ QYY @ alpha_y)
    ## --------------------------------
    # import solution
    # wx,wy = solution.CCA_HD(X,Y)
    ## --------------------------------
    return  centered_X @ alpha_x, centered_Y @ alpha_y
```

We now call the function we have implemented with a training sequence of 100 pairs of images. Because the returned solution is of same dimensions as the inputs, it can be rendered in a similar fashion.

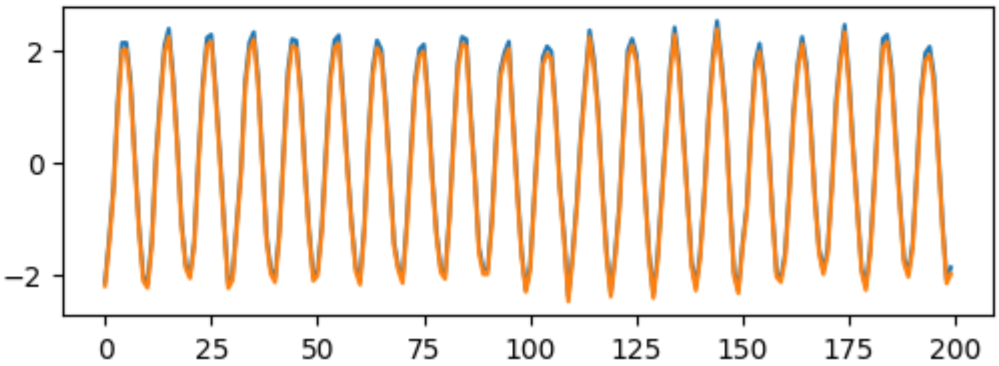
In [7]:
```
wx,wy = CCA_HD(X,Y)

utils.plotHDdata(wx,wy)
plt.show()
```




Here, we can clearly see a common factor that has been extracted between the two fields, specifically a point source emitting at a particular frequency. The sequence of image pairs can now be projected on these two filters:

In [8]:
```
plt.figure(figsize=(6,2))
plt.plot(wx.dot(X))
plt.plot(wy.dot(Y))
plt.show()
```




In [ ]: