Assignment 11

Friday, 17 January 2020 09:00

EMANUELE BELLI 03720425
THOMAS BRUNNER 03675118
LUCA CORBUCCI 03726968

Exercise 1

The representation of Leslie in the original space is

$$L = [0, 3, 0, 0, 4]$$

To find the representation of Leslie in the concept space we can do the following product:

$$L \cdot V = \begin{bmatrix} 0 & 3 & 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{bmatrix} = \begin{bmatrix} 1.74 & 2.84 \end{bmatrix}$$

Given the representation in the concept space we can understand how Leslie will like the others films:

$$R = [1.74 \quad 2.84] = U\Sigma = A \cdot V^T$$

(A ↓ data matrix)

$$A = R V^T$$

$$\begin{bmatrix} 1.74 & 2.84 \end{bmatrix} \begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{bmatrix} = \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \\ 2.01 \\ 2.01 \end{bmatrix}$$

Exercise 2

As A is non-random and of full rank, it corresponds to a shift-rotation of the space. Then

$$x_i \sim N(\mu, \omega\omega^T + \phi) \quad \text{[like in the lecture, marginalized } z_i\text{]}$$

$$\Rightarrow A x_i \sim N(A\mu, A(\omega\omega^T + \phi)A^T)$$

$$= N(A\mu, A\omega\omega^T A^T + A\phi A^T)$$

If the MLE for μ, ω, ϕ are μ_x, ω_x, ϕ_x for x_i, by "comparing" the expression we get:

$$\mu_{Ax} = A\mu_x$$
$$\omega_{Ax} = A\omega_x$$
$$\phi_{Ax} = A\phi_x A^T$$

In case A is orthogonal the expression simplifies further, under the assumption that ϕ is a scaled identity

$$A\phi A^T = A\sigma^2 I A^T = \sigma^2 A A^T$$
$$= \sigma^2 I$$

(↓ A orthonormal)

Exercise 3

W.l.o.g. X is centered matrix:

ⓐ $X^t X = \Gamma \Lambda \Gamma^T$

$$\Leftrightarrow (\lambda X)^T(\lambda X) = \Gamma \Lambda \lambda^2 \Gamma^t$$

$$\Leftrightarrow y_1^t y_1 = \Gamma \Lambda \lambda^2 \Gamma^t$$

i.e. all the eigenvalues have been scaled up by λ^2 ⇒ we still preserve 70 % of variance.

ⓑ Some idea:

$$R^t X^t X R = R^t \Gamma \Lambda \Gamma^t R$$

$$\Leftrightarrow (XR)^t (XR) = (R^t\Gamma)\Lambda(R^t\Gamma)^t$$

i.e. we apply a rotation to the eigenvectors and get the same eigenvalues ⇒ PCA explains again 70% of variance

ⓒ this is a combination of ⓐ and ⓑ

ⓓ we cannot tell precisely, because each column is scaled by a different amount

ⓔ A shift does not affect the variance explained ⇒ PCA preserves 70 % of the variance

ⓕ A is of rank S ⇒ it maps X into a space of dimension S ⇒ S dimensions explain 100% of the variance.

Exercise 4

ⓐ • Compute the mean to center the matrix X:

$$\text{mean} = \left[\frac{4+2+4-2}{4}, \frac{3+1-1+1}{4}, \frac{2-2+2+2}{4}\right] = [2, 1, 1]$$

• Center the matrix X:

$$\tilde{X} = \begin{bmatrix} 2 & 2 & 1 \\ 0 & 0 & -3 \\ 2 & -2 & 1 \\ -4 & 0 & 1 \end{bmatrix}$$

• Compute the variance:

$$Var(x_1) = \frac{1}{4}(4+0+4+16) = 6$$
$$Var(x_2) = \frac{1}{4}(4+0+4+0) = 2$$
$$Var(x_3) = \frac{1}{4}(1+9+1+1) = 3$$

• Compute the covariance matrix:

$$Cov((2,0,2,-4),(2,0,-2,0)) = \frac{1}{4}(4+0-4+0) - 0 = 0$$
$$Cov((2,0,2,-4),(1,-3,1,1)) = \frac{1}{4}(2+0+2-4) - 0 = 0$$
$$Cov((2,0,-2,0),(1,-3,1,1)) = \frac{1}{4}(2+0-2+0) - 0 = 0$$

this is the covariance matrix

$$\Sigma_x = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

the covariance matrix is a diagonal matrix so the eigenvalues are $(6, 2, 3)$.
The Γ matrix is the identity matrix

$$\Gamma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \Lambda = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

ⓑ

$$\Gamma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \Lambda = \begin{pmatrix} \boxed{6} & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & \boxed{3} \end{pmatrix}$$

$$Y = \begin{pmatrix} 2 & 2 & 1 \\ 0 & 0 & -3 \\ 2 & -2 & 1 \\ -4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 0 & -3 \\ 2 & 0 & 1 \\ -4 & 0 & 1 \end{pmatrix}$$

fraction of variance is: $\sum \lambda_i = 11$

$$\frac{9}{11} = 0,81$$

ⓒ $x_5 \in \mathbb{R}^3$

PCA on data including x_5 must lead to the same PCA of ⓐ

$$X = \begin{bmatrix} 4 & 3 & 2 \\ 2 & 1 & -2 \\ 4 & -1 & 2 \\ -2 & 1 & 2 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 & 1 \\ 0 & 0 & -3 \\ 2 & -2 & 1 \\ -4 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 6 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

we want the same mean but now we have 5 data points:

$$\frac{1}{5}(4+2+4-2+2) = 2$$
$$\frac{1}{5}(3+1-1+1+1) = 1$$
$$\frac{1}{5}(2-2+2+2+1) = 1$$

$x_5 = [2, 1, 1]$ the new $\tilde{X}$ is

$$\tilde{X} = \begin{bmatrix} 2 & 2 & 1 \\ 0 & 0 & -3 \\ 2 & -2 & 1 \\ -4 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$Var(x_1) = \frac{1}{5}(4+0+4+16+0) = \frac{24}{5}$$
$$Var(x_2) = \frac{1}{5}(4+0+4+0+0) = \frac{8}{5}$$
$$Var(x_3) = \frac{1}{5}(1+9+1+1+0) = \frac{12}{5}$$

$$\Sigma_x = \begin{bmatrix} \frac{24}{5} & 0 & 0 \\ 0 & \frac{8}{5} & 0 \\ 0 & 0 & \frac{12}{5} \end{bmatrix}$$

If we compute the reduced matrix Y we obtain the same matrix because in Σ_x the top 2 principal components are $\frac{24}{5}$ and $\frac{12}{5}$:

$$Y = \begin{pmatrix} 2 & 2 & 1 \\ 0 & 0 & -3 \\ 2 & -2 & 1 \\ -4 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 0 & -3 \\ 2 & 0 & 1 \\ -4 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

exercise_11_notebook

January 19, 2020

1 Programming task 11: Dimensionality Reduction

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt

        %matplotlib inline
```

1.1 Exporting the results to PDF

Once you complete the assignments, export the entire notebook as PDF and attach it to your homework solutions. The best way of doing that is 1. Run all the cells of the notebook. 2. Export/download the notebook as PDF (File -> Download as -> PDF via LaTeX (.pdf)). 3. Concatenate your solutions for other tasks with the output of Step 2. On a Linux machine you can simply use `pdfunite`, there are similar tools for other platforms too. You can only upload a single PDF file to Moodle.

Make sure you are using `nbconvert` Version 5.5 or later by running `jupyter nbconvert --version`. Older versions clip lines that exceed page width, which makes your code harder to grade.

1.2 PCA

Given the data in the matrix X your tasks is to: * Calculate the covariance matrix Σ. * Calculate eigenvalues and eigenvectors of Σ. * Plot the original data X and the eigenvectors to a single diagram. What do you observe? Which eigenvector corresponds to the smallest eigenvalue? * Determine the smallest eigenvalue and remove its corresponding eigenvector. The remaining eigenvector is the basis of a new subspace. * Transform all vectors in X in this new subspace by expressing all vectors in X in this new basis.

1.2.1 The given data X

```
In [2]: X = np.array([(-3,-2),(-2,-1),(-1,0),(0,1),
                      (1,2),(2,3),(-2,-2),(-1,-1),
                      (0,0),(1,1),(2,2), (-2,-3),
                      (-1,-2),(0,-1),(1,0), (2,1),(3,2)])
```

1.2.2 Task 1: Calculate the covariance matrix Σ

```
In [3]: def get_covariance(X):
            """Calculates the covariance matrix of the input data.

            Parameters
            ----------
            X : array, shape [N, D]
                Data matrix.

            Returns
            -------
            Sigma : array, shape [D, D]
                Covariance matrix

            """
            # TODO
            mean = np.mean(X, axis = 0)
            N = len(X)
            cov = (np.dot(X.T, X)-mean)/N

            # The covariance computed with this method is different than the covariance comput
            # np.cov() because we divide by N and numpy divide by N-1.
            return cov
```

1.2.3 Task 2: Calculate eigenvalues and eigenvectors of Σ.

```
In [4]: def get_eigen(S):
            """Calculates the eigenvalues and eigenvectors of the input matrix.

            Parameters
            ----------
            S : array, shape [D, D]
                Square symmetric positive definite matrix.

            Returns
            -------
            L : array, shape [D]
                Eigenvalues of S
            U : array, shape [D, D]
                Eigenvectors of S

            """
            # TODO
            eigenvalues, eigenvectors = np.linalg.eig(S)
            return eigenvalues, eigenvectors
```

1.2.4 Task 3: Plot the original data X and the eigenvectors to a single diagram.

Note that, in general if u_i is an eigenvector of the matrix M with eigenvalue λ_i then $\alpha \cdot u_i$ is also an eigenvector of M with the same eigenvalue λ_i, where α is an arbitrary scalar (including $\alpha = -1$).

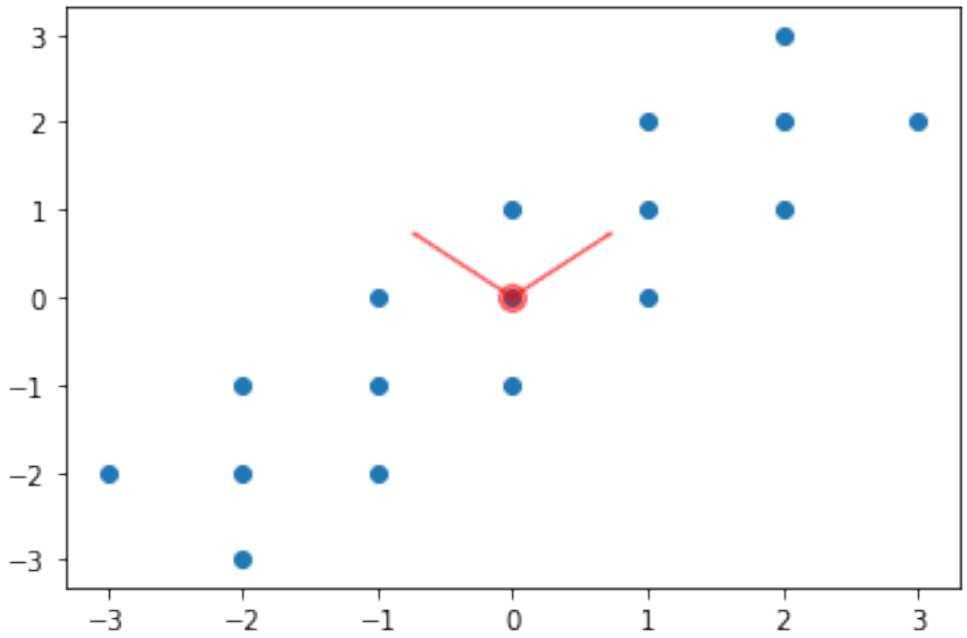
Thus, the signs of the eigenvectors are arbitrary, and you can flip them without changing the meaning of the result. Only their direction matters. The particular result depends on the algorithm used to find them.

```
In [5]: # plot the original data
        plt.scatter(X[:, 0], X[:, 1])

        # plot the mean of the data
        mean_d1, mean_d2 = X.mean(0)
        plt.plot(mean_d1, mean_d2, 'o', markersize=10, color='red', alpha=0.5)

        # calculate the covariance matrix
        Sigma = get_covariance(X)
        # calculate the eigenvector and eigenvalues of Sigma
        L, U = get_eigen(Sigma)

        plt.arrow(mean_d1, mean_d2, U[0, 0], U[1, 0], width=0.01, color='red', alpha=0.5)
        plt.arrow(mean_d1, mean_d2, U[0, 1], U[1, 1], width=0.01, color='red', alpha=0.5);
```



What do you observe in the above plot? Which eigenvector corresponds to the smallest eigenvalue?

Write your answer here:

[YOUR ANSWER]

1.2.5 Task 4: Transform the data

Determine the smallest eigenvalue and remove its corresponding eigenvector. The remaining eigenvector is the basis of a new subspace. Transform all vectors in X in this new subspace by expressing all vectors in X in this new basis.

```
In [6]: def transform(X, U, L):
            """Transforms the data in the new subspace spanned by the eigenvector correspondin

            Parameters
            ----------
            X : array, shape [N, D]
                Data matrix.
            L : array, shape [D]
                Eigenvalues of Sigma_X
            U : array, shape [D, D]
                Eigenvectors of Sigma_X

            Returns
            -------
            X_t : array, shape [N, 1]
                Transformed data

            """
            # TODO
            i = np.argmin(L)
            U[:,i] = 0
            temp_matrix = np.dot(X,U)
            X_t = np.delete(temp_matrix, i, axis = 1)

            return X_t
```

```
In [7]: X_t = transform(X, U, L)
```

1.3 SVD

1.3.1 Task 5: Given the matrix M find its SVD decomposition $M = U \cdot \Sigma \cdot V$ and reduce it to one dimension using the approach described in the lecture.

```
In [8]: M = np.array([[1, 2], [6, 3],[0, 2]])
```

```
In [9]: def reduce_to_one_dimension(M):
            """Reduces the input matrix to one dimension using its SVD decomposition.

            Parameters
            ----------
            M : array, shape [N, D]
                Input matrix.
```

```
        Returns
        -------
        M_t: array, shape [N, 1]
            Reduce matrix.

        """
        # TODO

        # Computation of S
        U, S, V = np.linalg.svd(M, full_matrices=False)
        n = len(S)
        i = np.argmin(S)
        S[i] = 0

        temp_matrix = U[:,:n] @ np.diag(S) @ V
        M_t = np.delete(temp_matrix, i, axis = 1)
        return M_t
```

In [10]:
```
M_t = reduce_to_one_dimension(M)
```