UML2015
Solutions and other comments
to some exercises from sets 1, 2 and 3

Set 1 Exercise 2:
The given task was simple, but also notice that induction implies the formula

$$\begin{aligned}\frac{d}{dt}\Big(A(t)B(t)\cdots Y(t)Z(t)\Big) &= \frac{dA(t)}{dt}B(t)\cdots Y(t)Z(t)\\ &+ A(t)\frac{dB(t)}{dt}\cdots Y(t)Z(t)\\ &+\cdots+ \\ &+ A(t)B(t)\cdots\frac{dY(t)}{dt}Z(t)\\ &+ A(t)B(t)\cdots Y(t)\frac{dZ(t)}{dt}\end{aligned} \tag{1}$$

(this symbolizes an arbitrary number of matrices, not the precise amount letters in our alphabet…) Simply write the product as

$$A(t)\Big(B(t)\cdots Y(t)Z(t)\Big) \tag{2}$$

and assume the result known for a smaller amount of matrices.

Also notice that the same idea will work when computing a partial derivative $\frac{\partial}{\partial A_{ij}}$. For example, suppose we want to compute $\frac{\partial f(A)}{\partial A_{ij}}$ for some f. Fix matrix A and indices i, j, denote $\overline{A}(0) = A$, and then define $\overline{A}(t)$ for all t with formula

$$\begin{cases} \overline{A}(t)_{i'j'} = A_{i'j'} & (i',j') \neq (i,j) \\ \overline{A}(t)_{i'j'} = A_{ij} + t & (i',j') = (i,j) \end{cases} \tag{3}$$

Now we have

$$\frac{df(\overline{A}(t))}{dt}\Big|_{t=0} = \frac{\partial f(A)}{\partial A_{ij}} \tag{4}$$

and we can use the product rule of differentiation if needed.

Set 1 Exercise 3:

$$\begin{aligned}\frac{\partial f(A)}{\partial A_{ij}} &= \frac{d}{dA_{ij}}(\mathbf{v}^T A^T C A \mathbf{v}) = \mathbf{v}^T \frac{dA^T}{dA_{ij}} C A \mathbf{v} + \mathbf{v}^T A^T C \frac{dA}{dA_{ij}}\mathbf{v} \\ &= \mathbf{v}_j (CA\mathbf{v})_i + (\mathbf{v}^T A^T C)_i \mathbf{v}_j \\ &= 2(CA\mathbf{v})_i \mathbf{v}_j\end{aligned} \tag{5}$$

The result can also be written in the form

$$\nabla_A f(A) = 2CA\mathbf{v}\mathbf{v}^T \tag{6}$$

since $(2CA\mathbf{v}\mathbf{v}^T)_{ij} = 2(CA\mathbf{v})_i\mathbf{v}_j$.

Set 1 Exercise 5:
The answers are

$$x_n = \alpha^{n-1}x_1 \quad \text{and} \quad y_n = \alpha^{2^{n-1}-1}y_1^{2^{n-1}} \tag{7}$$

$x_n \to 0$ obviously holds iff $|\alpha| < 1$

$y_n \to 0$ is slightly more complicated. We can write $y_n = \frac{1}{\alpha}(\alpha y_1)^{2^{n-1}}$, and now we see that $y_n \to 0$ holds iff $|\alpha y_1| < 1$. Hence the value of α alone does not determine the convergence, but the starting point y_1 plays a role too.

Set 2 Exercise 1:
1.1

$$f_1(x,y) = (x^2 - y)^2 \qquad \frac{\partial f_1(x,y)}{\partial x} = 4(x^2 - y)x \tag{8}$$

$$x_{n+1} = x_n - 4\mu(x_n^2 - y)x_n \tag{9}$$

We substitute $x_n = \sqrt{y} + \varepsilon_n$

$$\begin{aligned} \sqrt{y} + \varepsilon_{n+1} &= \sqrt{y} + \varepsilon_n - 4\mu\big((\sqrt{y} + \varepsilon_n)^2 - y\big)(\sqrt{y} + \varepsilon_n) \\ \implies \varepsilon_{n+1} &= (1 - 8\mu y)\varepsilon_n + O(\varepsilon_n^2) \end{aligned} \tag{10}$$

We see that $\varepsilon_{n+1} \approx (1 - 8\mu y)\varepsilon_n$ holds with small ε_n, so we can decide (avoiding all rigor) that the convergence is exponential if $0 < |1 - 8\mu y| < 1$.
1.2

$$f_2(x,y) = x^2 - y \qquad \frac{\partial f_2(x,y)}{\partial x} = 2x \tag{11}$$

$$x_{n+1} = x_n - \frac{x_n^2 - y}{2x_n} \tag{12}$$

We substitute $x_n = \sqrt{y} + \varepsilon_n$

$$\begin{aligned} \sqrt{y} + \varepsilon_{n+1} &= \sqrt{y} + \varepsilon_n - \frac{(\sqrt{y} + \varepsilon_n)^2 - y}{2(\sqrt{y} + \varepsilon_n)} \\ \implies \varepsilon_{n+1} &= \frac{1}{2\sqrt{y}}\varepsilon_n^2 + O(\varepsilon_n^3) \end{aligned} \tag{13}$$

We see that $\varepsilon_{n+1} \approx \frac{1}{2\sqrt{y}}\varepsilon_n^2$ holds with small ε_n, so we can decide that the convergence is faster than exponential.

If we set $\mu = \frac{1}{8y}$ in 1.1, we can make it faster than exponential too.

Set 2 Exercise 2:

2.1

We fix index i, then use the definition of the partial derivative (Equation (4) in the exercise sheet), and then use the given assumption (Equation (5) in the exercise sheet) with a vector $\mathbf{h} = \mathbf{e}_i$.

$$\frac{\partial J(\mathbf{x})}{\partial x_i} = \lim_{\varepsilon \to 0} \frac{\overbrace{J(\mathbf{x} + \varepsilon \mathbf{e}_i)}^{=J(\mathbf{x})+\varepsilon \mathbf{v}\cdot\mathbf{e}_i + o(\varepsilon)} - J(\mathbf{x})}{\varepsilon} = \lim_{\varepsilon \to 0} \Big(\mathbf{v} \cdot \mathbf{e}_i + o(1) \Big) = \mathbf{v}_i \tag{14}$$

Hence $\nabla J(\mathbf{x}) = \mathbf{v}$.

2.2

We fix indices i, j, then use the definition of the partial derivative (Equation (6) in the exercise sheet), and then use the given assumption (Equation (7) in the exercise sheet) with vectors $\mathbf{h} = \mathbf{e}_i$ and $\mathbf{w} = \mathbf{e}_j$.

$$\frac{\partial J(A)}{\partial A_{ij}} = \lim_{\varepsilon \to 0} \frac{\overbrace{J(A + \varepsilon \mathbf{e}_i \mathbf{e}_j^T)}^{=J(A)+\varepsilon \mathbf{e}_i^T B \mathbf{e}_j + o(\varepsilon)} - J(A)}{\varepsilon} = \lim_{\varepsilon \to 0} \Big(\mathbf{e}_i^T B \mathbf{e}_j + o(1) \Big) = B_{ij} \tag{15}$$

Hence $\nabla f(A) = B$.

2.3

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & & \vdots \\ A_{N1} & \cdots & A_{NN} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & \cdots & B_{1N} \\ \vdots & & \vdots \\ B_{N1} & \cdots & B_{NN} \end{pmatrix} \tag{16}$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_{N^2} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{N^2} \end{pmatrix}, \tag{17}$$

$$A = \begin{pmatrix} x_1 & \cdots & x_N \\ x_{N+1} & \cdots & x_{2N} \\ \vdots & & \vdots \\ x_{N^2-N} & \cdots & x_{N^2} \end{pmatrix}, \quad B = \begin{pmatrix} y_1 & \cdots & y_N \\ y_{N+1} & \cdots & y_{2N} \\ \vdots & & \vdots \\ y_{N^2-N} & \cdots & y_{N^2} \end{pmatrix} \tag{18}$$

The requested result comes with simple calculation:

$$\mathrm{Tr}(B^TA) \underset{*}{=} \sum_{n=1}^{N}(B^TA)_{nn} \underset{**}{=} \sum_{n,n'=1}^{N} B_{n'n}A_{n'n} \underset{***}{=} \sum_{k=1}^{N^2} y_kx_k = \mathbf{y}^T\mathbf{x} \tag{19}$$

In step * we use the definition of trace. In step ** we use the definition of the matrix multiplication and the transpose, and then write the two sums over indices n and n' as a one sum over the index pair (n, n'). In the critical step *** we recognize that in the sum the index pair (n, n') goes through the all N^2 values from the set $\{1, 2, \dots, N\} \times \{1, 2, \dots, N\}$, and therefore the sum must result in the same value as the sum over all N^2 products y_kx_k. Here it is obvious that it doesn't matter which way the matrix elements have been ordered into the vector form, because the sum goes over all the same elements anyway. It is important though that the elements have been ordered in the same way in both A and B.

2.4

First we have a function $J : \mathbb{R}^{N\times N} \to \mathbb{R}$. We can define a new function $f : \mathbb{R}^{N^2} \to \mathbb{R}$ which produces the same values once the input parameters have been identified with some ordering. In other words $J(A) = f(\mathbf{x})$ when $A \sim \mathbf{x}$ in the same spirit as above in 2.3. Let W be a fixed $N \times N$ matrix. We choose a vector $\mathbf{w}$ so that $W \sim \mathbf{w}$ in the same way as $A \sim \mathbf{x}$. From vector calculus we know

$$f(\mathbf{x} + \varepsilon\mathbf{w}) = f(\mathbf{x}) + \varepsilon\mathbf{w} \cdot \nabla f(\mathbf{x}) + o(\varepsilon). \tag{20}$$

Next we need to justify that also $\nabla J(A) \sim \nabla f(\mathbf{x})$. Then, by the previous exercise, we have

$$\mathbf{w} \cdot \nabla f(\mathbf{x}) = \mathrm{Tr}(W^T\nabla J(A)) \tag{21}$$

and the claimed result

$$J(A + \varepsilon W) = J(A) + \varepsilon\mathrm{Tr}(W^T\nabla J(A)) + o(\varepsilon) \tag{22}$$

is done. Justifying $\nabla J(A) \sim \nabla f(\mathbf{x})$ is clear, because in the definition of the partial derivatives, a matrix $\mathbf{e}_i\mathbf{e}_j^T$ with some $1 \le i, j, \le N$ is equivalent with a vector $\mathbf{e}_k$ with some $1 \le k \le N^2$. The elements of $\nabla J(A)$ become ordered into a vector $\nabla f(\mathbf{x})$ just like the elements of A are ordered into a vector $\mathbf{x}$.

If we substitute $W = \mathbf{h}\mathbf{w}^T$, we get

$$\mathrm{Tr}(W^T\nabla J(A)) = \mathrm{Tr}(\mathbf{w}\mathbf{h}^T\nabla J(A)) = \mathrm{Tr}(\mathbf{h}^T\nabla J(A)\mathbf{w}) = \mathbf{h}^T\nabla J(A)\mathbf{w} \tag{23}$$

so the previous results are compatible with each other.

Some lessons to consider: The point is that an $N \times K$ matrix can always be seen as an NK component vector, in other words as an $NK \times 1$ matrix. This alone looks simple, but the implications are not necessarily obvious to all at first sight. For example, suppose we want to find some complex matrix $\overline{A} \in \mathbb{C}^{N\times N}$.

Then suppose that we have succeeded in defining a twice differentiable function $f : \mathbb{C}^{N\times N} \to \mathbb{R}$ such that the seeked $\overline{A}$ is its maximum. Then we define a sequence of complex matrices $(A(k))_{k=0,1,2,\ldots}$ by the recursion formulas

$$\begin{aligned} \mathrm{Re}(A(k+1)_{nn'}) &= \mathrm{Re}(A(k)_{nn'}) + \mu \frac{\partial f(A(k))}{\partial \mathrm{Re}(A_{nn'})} \\ \mathrm{Im}(A(k+1)_{nn'}) &= \mathrm{Im}(A(k)_{nn'}) + \mu \frac{\partial f(A(k))}{\partial \mathrm{Im}(A_{nn'})} \end{aligned} \tag{24}$$

In the end the complex matrix is an $N^2 \times 1$ complex vector, and that in turn is a $2N^2 \times 1$ real vector, so the graph of the function f near the maximum is roughly a paraboloid in $2N^2 + 1$ dimensions. Since it is a paraboloid, with proper μ the matrix sequence will convergence to $\overline{A}$. Hence, even complex matrices can be computed like this.

Set 2 Exercise 4:
4.2

We must prove $AU = U\Lambda$, when the columns of U are linearly independent eigenvectors of A, i.e. $U_{*i} = \mathbf{u}_i$, $U_{ji} = (\mathbf{u}_i)_j$ and $A\mathbf{u}_i = \lambda_i \mathbf{u}_i$, and Λ is a diagonal matrix with diagonal elements $\Lambda_{ii} = \lambda_i$.

The formula $AU = U\Lambda$ is so obvious that it can be justified in many roughly equivalent ways, and there is no one right solution. We go through a reasonably detailed proof next, and first ask that how do we prove

$$(AU)_{*i} = A(U_{*i})? \tag{25}$$

Here $(AU)_{*i}$ means that we first multiply two matrices, and then take a restriction to the column i. The notation $A(U_{*i})$ instead means that we first form a vector U_{*i} by a restriction to one column of U, and then apply the multiplication of a matrix and a vector. Two vectors are the same if their elements are the same, so we examine the j:th element of left and right side separately. The left side is

$$(AU)_{ji} = \sum_{k=1}^{n} A_{jk} U_{ki} \tag{26}$$

by the definition of multiplication of two matrices, and the right side is

$$\big(A(U_{*i})\big)_j = \sum_{k=1}^{N} A_{jk} \big(U_{*i}\big)_k = \sum_{k=1}^{n} A_{jk} U_{ki} \tag{27}$$

by the definition of multiplication of matrix and a vector. The left and right side had the same elements, so $(AU)_{*i} = A(U_{*i})$ is right. Then

$$(AU)_{*i} = A(U_{*i}) = A\mathbf{u}_i = \lambda_i \mathbf{u}_i = \Lambda_{ii} U_{*i} \tag{28}$$

$$\implies \quad (AU)_{ji} = \Lambda_{ii} U_{ji} \tag{29}$$

From the other direction

$$(U\Lambda)_{ji} = \sum_{k=1}^{n} U_{jk} \underbrace{\Lambda_{ki}}_{=\delta_{ki}\Lambda_{ii}} = U_{ji}\Lambda_{ii} \tag{30}$$

The elements of the matrices AU and $U\Lambda$ are the same, so the matrices are the same.

4.3 (half) We know $AU = U\Lambda$, and we denote $V = (U^{-1})^T$. The formula

$$A = U\Lambda V^T \tag{31}$$

is obtained by simply multiplying the both sides of $AU = U\Lambda$ with V^T from right, and using $UV^T = \text{id}$. The the columns of V are denoted as $\mathbf{v}_i$, i.e. $V_{*i} = \mathbf{v}_i$ and $V_{ji} = (\mathbf{v}_i)_j$, and we are asked to prove

$$A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{v}_i^T. \tag{32}$$

We examine an arbitrary element A_{jk} by using the formula $A = U\Lambda V^T$.

$$\begin{aligned} A_{jk} = (U\Lambda V^T)_{jk} &= \sum_{i,i'=1}^{n} U_{ji} \underbrace{\Lambda_{ii'}}_{=\delta_{ii'}\Lambda_{ii}} (V^T)_{i'k} = \sum_{i=1}^{n} U_{ji}\Lambda_{ii}V_{ki} \\ &= \sum_{i=1}^{n} \lambda_i (\mathbf{u}_i)_j (\mathbf{v}_i)_k = \sum_{i=1}^{n} \lambda_i \big(\mathbf{u}_i \mathbf{v}_j^T\big)_{jk} = \Big(\sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{v}_j^T\Big)_{jk} \end{aligned} \tag{33}$$

The following lemma should be proven separately, if not known in advance. Suppose $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ are some vectors. How do you prove $\mathbf{x}_j \mathbf{y}_k = (\mathbf{x}\mathbf{y}^T)_{jk}$? This should be seen as a consequence of the matrix multiplication. $\mathbf{x}$ can be seen as an $n \times 1$ matrix, and $\mathbf{y}^T$ as a $1 \times n$ matrix, so their multiplication is

$$(\mathbf{x}\mathbf{y}^T)_{jk} = \sum_{\ell=1}^{1} \mathbf{x}_{j\ell}(\mathbf{y}^T)_{\ell k} = \mathbf{x}_{j,1}(\mathbf{y}^T)_{1,k} = \mathbf{x}_{j,1}\mathbf{y}_{k,1}. \tag{34}$$

Then we denote $\mathbf{x}_{j,1} = \mathbf{x}_j$ and $\mathbf{y}_{k,1} = \mathbf{y}_k$.

Set 3 Exercise 1:
1.3
$g = f \circ I$, so by the ordinary chain rule:

$$\frac{\partial g(A)}{\partial A_{ij}} = \sum_{n,n'=1}^{N} \frac{\partial f(I(A))}{\partial I_{nn'}} \frac{\partial I_{nn'}(A)}{\partial A_{ij}} \tag{35}$$

From 1.2 we know

$$\frac{\partial I_{nn'}(A)}{\partial A_{ij}} = -(A^{-1})_{ni}(A^{-1})_{jn'} \tag{36}$$

By substituting this, we get a formula which only involves the gradient of f and the matrix A^{-1}, which we can assume to be available. The end result can also be written as

$$\nabla g(A) = -(A^{-1})^T \nabla f(A^{-1})(A^{-1})^T \tag{37}$$

With this kind of results always check that the result simplifies to a known result in the special case $N = 1$. If $f : \mathbb{R} \to \mathbb{R}$ is some differentiable function,

$$\frac{d}{da} f\Big(\frac{1}{a}\Big) = -\frac{1}{a^2} f'\Big(\frac{1}{a}\Big) \tag{38}$$

looks right. Some confusion can arise from the problem that it is perhaps not obvious if we should denote the partial derivatives of f as

$$\frac{\partial f}{\partial I_{nm}}, \quad \frac{\partial f}{\partial A_{nm}}, \quad \text{or} \quad \frac{\partial f}{\partial (A^{-1})_{nm}}. \tag{39}$$

These are all supposed to be notation for the same functions (assuming we know what we are doing). A standard convention is that the partial derivative is written with respect to that variable which we "usually" substitute into f, or which we used in some "original" definition of f.

Set 3 Exercise 2:
2.2
The calculation required the following trick:

$$\begin{aligned}
\frac{\partial}{\partial W_{ij}}(\mathbf{v}_n^T W \mathbf{u}_n) &= \Big(\frac{\partial}{\partial W_{ij}}\mathbf{v}_n^T\Big) \underbrace{W\mathbf{u}_n}_{=\lambda_n \mathbf{u}_n} + \mathbf{v}_n^T \underbrace{\Big(\frac{\partial}{\partial W_{ij}} W\Big)}_{=\mathbf{e}_i \mathbf{e}_j^T} \mathbf{u}_n + \underbrace{\mathbf{v}_n^T W}_{=\lambda_n \mathbf{v}_n^T} \Big(\frac{\partial}{\partial W_{ij}} \mathbf{u}_n\Big) \\
&= \lambda_n \underbrace{\Big(\Big(\frac{\partial}{\partial W_{ij}}\mathbf{v}_n^T\Big)\mathbf{u}_n + \mathbf{v}_n^T\Big(\frac{\partial}{\partial W_{ij}}\mathbf{u}_n\Big)\Big)}_{=\frac{\partial}{\partial W_{ij}}(\mathbf{v}_n^T \mathbf{u}_n) = 0} + (\mathbf{v}_n)_i (\mathbf{u}_n)_j \\
&= (\mathbf{v}_n \mathbf{u}_n^T)_{ij}
\end{aligned} \tag{40}$$

Set 3 Exercise 3:
3.1

$$\ell(\boldsymbol{\mu}, \Sigma) = -\frac{K}{2} \log\big(\det(\Sigma)\big) - \frac{1}{2}\sum_{k=1}^{K} (\mathbf{x}_k - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) + \text{const.} \tag{41}$$

3.2

$$
\begin{aligned}
\frac{\partial \ell}{\partial \mu_n} &= -\frac{1}{2}\sum_{k=1}^{K}\Big(\big(\frac{\partial}{\partial \mu_n}(\mathbf{x}_k-\boldsymbol{\mu})^T\big)\Sigma^{-1}(\mathbf{x}_k-\boldsymbol{\mu}) + (\mathbf{x}_k-\boldsymbol{\mu})^T\Sigma^{-1}\big(\frac{\partial}{\partial \mu_n}(\mathbf{x}_k-\boldsymbol{\mu})\big)\Big)\\
&= -\frac{1}{2}\sum_{k=1}^{K}\Big(-(\Sigma^{-1})_{n*}(\mathbf{x}_k-\boldsymbol{\mu}) - (\mathbf{x}_k-\boldsymbol{\mu})^T(\Sigma^{-1})_{*n}\Big)\\
&= \sum_{k=1}^{K}\big(\Sigma^{-1}(\mathbf{x}_k-\boldsymbol{\mu})\big)_n\\
&= \Big(\Sigma^{-1}\Big(\sum_{k=1}^{K}\mathbf{x}_k - K\boldsymbol{\mu}\Big)\Big)_n
\end{aligned}
\tag{42}
$$

$$
\nabla_{\boldsymbol{\mu}}\ell = \Sigma^{-1}\Big(\sum_{k=1}^{K}\mathbf{x}_k - K\boldsymbol{\mu}\Big) \tag{43}
$$

$$
\begin{aligned}
\frac{\partial \ell}{\partial \Sigma_{nn'}} &= -\frac{K}{2}(\Sigma^{-1})_{nn'} - \frac{1}{2}\sum_{k=1}^{K}\underbrace{(\mathbf{x}_k-\boldsymbol{\mu})^T(-1)(\Sigma^{-1})_{*n}(\Sigma^{-1})_{n'*}(\mathbf{x}_k-\boldsymbol{\mu})}_{=-\left(\Sigma^{-1}(\mathbf{x}_k-\boldsymbol{\mu})\right)_{n*}\left((\mathbf{x}_k-\boldsymbol{\mu})^T\Sigma^{-1}\right)_{*n'}}\\
&= -\frac{K}{2}(\Sigma^{-1})_{nn'} + \frac{1}{2}\Big(\Sigma^{-1}\Big(\sum_{k=1}^{K}(\mathbf{x}_k-\boldsymbol{\mu})(\mathbf{x}_k-\boldsymbol{\mu})^T\Big)\Sigma^{-1}\Big)_{nn'}
\end{aligned}
\tag{44}
$$

$$
\nabla_{\Sigma}\ell = -\frac{K}{2}\Sigma^{-1} + \frac{1}{2}\Sigma^{-1}\Big(\sum_{k=1}^{K}(\mathbf{x}_k-\boldsymbol{\mu})(\mathbf{x}_k-\boldsymbol{\mu})^T\Big)\Sigma^{-1} \tag{45}
$$

Set 3 Exercise 4:
4.1

The function $J(\mathbf{w})$ essentially involves the sum of squared elements of the $N \times K$ matrix $X - \hat{X}$, and therefore it can be written as

$$
J(\mathbf{w}) = \frac{1}{K}\mathrm{Tr}\big((X^T - \hat{X}^T)(X - \hat{X})\big) \tag{46}
$$

$\hat{X} = \mathbf{w}Z$ and $Z = \mathbf{w}^T X$, so $\hat{X} = \mathbf{w}\mathbf{w}^T X$, and then $X - \hat{X} = (\mathrm{id} - \mathbf{w}\mathbf{w}^T)X$.

By using the properties of trace and $\mathbf{w}^T\mathbf{w} = 1$, we get

$$\begin{aligned} J(\mathbf{w}) &= \frac{1}{K}\mathrm{Tr}\big(X^T \underbrace{(\mathrm{id} - \mathbf{w}\mathbf{w}^T)(\mathrm{id} - \mathbf{w}\mathbf{w}^T)}_{=\mathrm{id}-\mathbf{w}\mathbf{w}^T} X\big) \\ &= \frac{1}{K}\mathrm{Tr}\big((\mathrm{id} - \mathbf{w}\mathbf{w}^T)XX^T\big) \\ &= \frac{1}{K}\mathrm{Tr}(XX^T) - \frac{1}{K}\mathbf{w}^T XX^T\mathbf{w} \end{aligned} \tag{47}$$

4.2

One possibility:

We are interested in maximizing

$$\lambda_1 m_1 + \cdots + \lambda_N m_N \tag{48}$$

with respect to $(m_1, \ldots, m_N)$ under the constraints $0 \leq m_n$ for all $1 \leq n \leq N$ and $m_1 + \cdots + m_N = 1$. By substituting $m_1 = 1 - m_2 - \cdots - m_N$ we can equivalently study maximizing

$$\lambda_1 + (\lambda_2 - \lambda_1)m_2 + \cdots (\lambda_N - \lambda_1)m_N \tag{49}$$

with respect to $(m_2, \ldots, m_N)$ under the constraints $0 \leq m_n$ for all $2 \leq n \leq N$ and $m_2 + \cdots + m_N \leq 1$. We see that the point $(m_2, \ldots, m_N) = (0, \ldots, 0)$ is in the allowed domain, and there the objective quantity assumes the value λ_1. On the other hand $(\lambda_n - \lambda_1)m_n \leq 0$ for all $2 \leq n \leq N$, and therefore it is not possible for the objective quantity to assume values greater than λ_1. Also, if $m_n \neq 0$ with some $2 \leq n \leq N$, then the objective quantity necessarily assumes a value less than λ_1.

Second possibility:

We want to prove that

$$\lambda_1 m_1 + \cdots + \lambda_N m_N < \lambda_1 \tag{50}$$

holds if $(m_1, m_2, \ldots, m_N) \neq (1, 0, \ldots, 0)$ (under the other assumptions mentioned in the exercise sheet). So we assume $m_1 \neq 0$. Then

$$\frac{m_2}{1 - m_1} + \cdots + \frac{m_N}{1 - m_1} = 1 \tag{51}$$

We can make an induction assumption that the result is already known with $N - 1$ coefficients. Therefore we know that necessarily

$$\frac{\lambda_2 m_2}{1 - m_1} + \cdots + \frac{\lambda_N m_N}{1 - m_1} \leq \lambda_2 \tag{52}$$

On the other hand $\lambda_2 < \lambda_1$, so we now know

$$\begin{aligned} &\frac{\lambda_2 m_2}{1-m_1} + \cdots + \frac{\lambda_N m_N}{1-m_1} < \lambda_1 \\ \implies &\lambda_2 m_2 + \cdots + \lambda_N m_N < (1-m_1)\lambda_1 \\ \implies &\lambda_1 m_1 + \cdots + \lambda_N m_N < \lambda_1 \end{aligned} \tag{53}$$

4.3

We assume that A has the size $N \times N$. Then let U be a matrix whose columns U_{*n} are independent eigenvectors of A, ordered so that the largest eigenvalues λ_n correspond to smallest n. We relate vectors $\mathbf{w}$ and $\mathbf{v}$ by relation $\mathbf{w} = U\mathbf{v}$ (and $\mathbf{v} = U^T\mathbf{w}$). $\|\mathbf{w}\| = \|\mathbf{v}\|$ because U is orthogonal.

$$\mathbf{w}^T A\mathbf{w} = \mathbf{v}^T U^T A U\mathbf{v} = \sum_{n=1}^{N} \lambda_n v_n^2 \tag{54}$$

From the previous exercise we know that this will be maximized with respect to $\mathbf{v}$ by the choice $(v_1^2, v_2^2, \ldots, v_N^2) = (1, 0 \ldots, 0)$ under the constraint $\|\mathbf{v}\| = 1$. In other words $v_1 = \pm 1$. Then $\mathbf{w} = U\mathbf{v} = \pm U_{*1}$, so we see that $\mathbf{w}$ must be an eigenvector corresponding to the largest eigenvalue.