EQ2415 – Machine Learning and Data Science
HT22

Tutorial 1 A. Honoré, A. Ghosh

1 Inference in linear models

1.1 Projection on a line.

The set $\mathcal{L} = \{\beta \mathbf{u} \mid \beta \in \mathbb{R}\}$ where $\mathbf{u} \in \mathbb{R}^d$ is a unit vector, defines a line of points that may be obtained by varying the value of β.

Question 1. Derive an expression for the point $\mathbf{y}$ that lies on this line $\mathcal{L}$, and that is as close as possible (according to the Euclidean distance) to an arbitrary point $\mathbf{x} \in \mathbb{R}^d$. This operation of replacing a point by its nearest member within some set is called projection.

Solution: We begin by defining the distance from $\mathbf{y}$ to $\mathbf{x}$. We would like to find the $\mathbf{y}$ that minimizes this distance:

$$||\mathbf{x} - \mathbf{y}||^2. \tag{1}$$

Next, we need to enforce the constraint that $\mathbf{y}$ lies on the line defined by $\mathcal{L}$. We can do this simply by defining $\mathbf{y} = \alpha \mathbf{u}$ for some $\alpha \in \mathbb{R}$.

$$||\mathbf{x} - \alpha\mathbf{u}||^2. \tag{2}$$

Next, we expand the expression:

$$\begin{aligned} l(\alpha) &= ||\mathbf{x} - \alpha\mathbf{u}||^2 \\ &= (\mathbf{x} - \alpha\mathbf{u})^T(\mathbf{x} - \alpha\mathbf{u}) \\ &= \mathbf{x}^T\mathbf{x} - 2\alpha\mathbf{x}^T\mathbf{u} + \alpha^2\mathbf{u}^T\mathbf{u} \\ &= \mathbf{x}^T\mathbf{x} - 2\alpha\mathbf{x}^T\mathbf{u} + \alpha^2 \end{aligned} \tag{3}$$

In the last line, we used the fact that $\mathbf{u}$ is a unit vector (i.e. $\mathbf{u}^T\mathbf{u} = 1$) to make the simplification.

We can minimize this distance by taking the derivative with respect to α and setting it to zero:

$$\begin{aligned} &\frac{dl(\alpha)}{d\alpha} = 0 \\ \implies &-2\mathbf{x}^T\mathbf{u} + 2\alpha = 0 \\ \implies &\alpha = \mathbf{x}^T\mathbf{u}. \end{aligned} \tag{4}$$

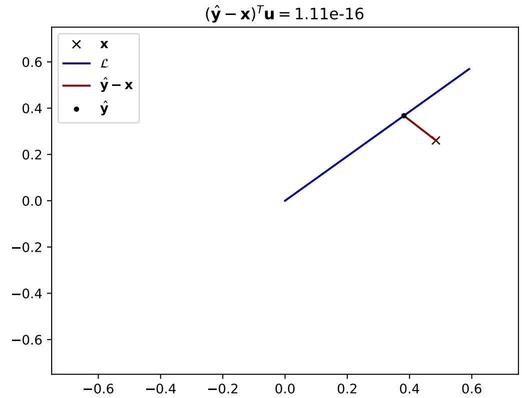
Recalling that $\mathbf{y} = \alpha\mathbf{u}$, we can conclude that $\mathbf{y} = (\mathbf{x}^T\mathbf{u})\mathbf{u}$. ■

Question 2. Write a small Python program that calculates the projection of random points $\mathbf{y}$ on the line generated by a unit vector $\mathbf{u}$. Use a space of dimension $d = 2$.

Solution:

```
x0=np.zeros(d); x1=np.random.rand(d)*0.75; x=(x1-x0).reshape(d,n);
u0=np.zeros(d); u1=np.random.rand(d)*0.75; u=(u1-u0).reshape(d,n); u=u/np.linalg.norm(u);
yhat = (x.T@u)*u;
fig, ax = plt.subplots() ax.plot(x[0,0], x[1,0],'x',label="x",color="black") ax.plot([u0[0], u1[0]],
[u0[1], u1[1]], label="ℒ",color="darkblue")
ax.plot([x[0,0], yhat[0,0]], [x[1,0], yhat[1,0]], label="ŷ − x",color="darkred")
ax.plot(yhat[0,0], yhat[1,0],'.',label="ŷ",color="black")
ax.legend() ax.set_title("(ŷ − x)ᵀu ="+":.2e".format( ((yhat-x).T @ u)[0,0]));
w=.75; ax.set_xlim([-w,w]); ax.set_ylim([-w,w]);
```

■



1.2 Some matrix algebra

Let $m, n > 0$.

Vector by scalar Suppose that a vector $\mathbf{y} \in \mathbb{R}^m$ is dependent upon a scalar $\alpha \in \mathbb{R}$. Then the derivative of $\mathbf{y}$ with respect to α is the vector:

$$J = \frac{\partial \mathbf{y}}{\partial \alpha} = \begin{bmatrix} \frac{dy_1}{d\alpha} \\ \vdots \\ \frac{dy_m}{d\alpha} \end{bmatrix} \tag{5}$$

Scalar by vector Suppose that a scalar $x \in \mathbb{R}$ depends upon a vector $\mathbf{y} \in \mathbb{R}^m$. Then the derivative of x with respect to $\mathbf{y}$ is the (row) vector:

$$J = \frac{\partial x}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial x}{\partial y_1} & \cdots & \frac{\partial x}{\partial y_m}. \end{bmatrix} \tag{6}$$

Vector by vector Suppose we have m real valued multivariate functions $f_i : \mathbb{R}^n \to \mathbb{R}$. Suppose also that we have a multivariate function $f : \mathbb{R}^n \to \mathbb{R}^m$ is such that for some $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$,

$$\mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})). \tag{7}$$

The Jacobian matrix J, of the multivariate function f, has elements

$$J_{ij} = \frac{\partial f_i(\mathbf{x})}{\partial x_j}, \text{ for } i = 1, \dots, m \text{ and } j = 1, \dots, n, \tag{8}$$

i.e. can be written in matrix form:

$$J = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix}. \tag{9}$$

Scalar by matrix Suppose a scalar $x \in \mathbb{R}$ is dependent upon a matrix $M \in \mathbb{R}^{m \times n}$. Then the derivative of x wrt that matrix is written in matrix form:

$$J = \frac{\partial x}{\partial M} = \begin{bmatrix} \frac{\partial x}{\partial M_{11}} & \cdots & \frac{\partial x}{\partial M_{1m}} \\ \vdots & & \vdots \\ \frac{\partial x}{\partial M_{n1}} & \cdots & \frac{\partial x}{\partial M_{nm}} \end{bmatrix} \tag{10}$$

Suppose that for $m, n > 0$, we have $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^m$ and $M \in \mathbb{R}^{n \times m}$.

Question 1. Calculate the Jacobian:

1. $\dfrac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} =$

2. $\dfrac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} =$

3. $\dfrac{\partial \mathbf{a}^T M \mathbf{b}}{\partial M} =$

4. $\dfrac{\partial \mathbf{b}^T M^T M \mathbf{c}}{\partial M} =$

5. $\dfrac{\partial ||\mathbf{x}||^2}{\partial \mathbf{x}} =$

Solution:

1. $\dfrac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}^T$

2. $\dfrac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}^T$

3. $\dfrac{\partial \mathbf{a}^T M \mathbf{b}}{\partial M} = \mathbf{b}\mathbf{a}^T$

4. $\dfrac{\partial \mathbf{b}^T M^T M \mathbf{c}}{\partial M} = M(\mathbf{b}\mathbf{c}^T + \mathbf{c}\mathbf{b}^T)$

5. $\dfrac{\partial ||\mathbf{x}||_2^2}{\partial \mathbf{x}} = 2\mathbf{x}^T$

■

1.3 Minimum mean square error

Suppose that we can observe two random variables $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^q$. Suppose also that these variables are related, and that we model this relation by a linear model parameterized with a matrix $A \in \mathbb{R}^{q \times d}$, i.e. such that

$$\mathbf{y} = A\mathbf{x}. \tag{11}$$

Question 1. Find $A^\star$ leading to the minimum mean square error, i.e. find $A^\star$ such that

$$A^\star = \arg\min_A \mathbb{E}[||\mathbf{y} - A\mathbf{x}||^2]. \tag{12}$$

Solution: We will find the value of A such that the derivative of the expectation in (12) is 0.

$$\begin{aligned}
\frac{\partial \mathbb{E}[(\mathbf{y} - A\mathbf{x})^T(\mathbf{y} - A\mathbf{x})]}{\partial A} &= \mathbb{E}\left[\frac{\partial \mathbf{x}^T A^T A \mathbf{x}}{\partial A} - \frac{\partial \mathbf{x}^T A^T \mathbf{y}}{\partial A} - \frac{\partial \mathbf{y}^T A \mathbf{x}}{\partial A} + \frac{\partial \mathbf{y}^T \mathbf{y}}{\partial A}\right] \\
&= \mathbb{E}[2A\mathbf{x}\mathbf{x}^T - \mathbf{x}\mathbf{y}^T - \mathbf{x}\mathbf{y}^T], \text{ since } \frac{\partial \mathbf{y}^T A \mathbf{x}}{\partial A} = \frac{\partial \mathbf{x}^T A^T \mathbf{y}}{\partial A} = \mathbf{x}\mathbf{y}^T \\
&= \mathbb{E}[2A\mathbf{x}\mathbf{x}^T - \mathbf{y}\mathbf{x}^T - \mathbf{y}\mathbf{x}^T], \text{ since } \mathbf{x}\mathbf{y}^T \text{ is symmetric} \\
&= 2\mathbb{E}\left[(A\mathbf{x} - \mathbf{y})\mathbf{x}^T\right]
\end{aligned} \tag{13}$$

Setting to 0 and rearranging (13):

$$\begin{aligned}
&\left.\frac{\partial \mathbb{E}[(\mathbf{y} - A\mathbf{x})^T(\mathbf{y} - A\mathbf{x})]}{\partial A}\right|_{A^\star} = 0 \\
&\implies A^\star \mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbb{E}[\mathbf{y}\mathbf{x}^T] \\
&\implies A^\star = \mathbb{E}[\mathbf{y}\mathbf{x}^T]\left(\mathbb{E}[\mathbf{x}\mathbf{x}^T]\right)^{-1}.
\end{aligned} \tag{14}$$

$\blacksquare$

Question 2. Ordinary Least squares.

Suppose that you are given n data points for $\mathbf{x}$ and $\mathbf{y}$, in the form of matrices $X \in \mathbb{R}^{d\times n}$ and $Y \in \mathbb{R}^{q\times n}$ respectively.

Find the least square solution $A_n^\star$ of the fitting problem when n samples are observed. Express the solution as a function of the matrices X and Y.

Solution: Remember that

$$\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{y}_i - A\mathbf{x}_i||^2 \to \mathbb{E}[||\mathbf{y} - A\mathbf{x}||^2], \text{ when } n \to \infty. \tag{15}$$

This means that the least square solution and the minimum mean square solution coincide when infinitely many samples are observed.

The least square solution $A_n^\star$, minimizes the Frobenius norm of the residuals, i.e.

$$\left.\frac{\partial ||Y - AX||_F^2}{\partial A}\right|_{A_n^\star} = 0 \\ \implies {}^\star \tag{16}$$

* **Frobenius norm derivative** The frobenius norm for a real valued matrix is sum of the square of the elements of the matrix, and can be written:

$$\begin{aligned} ||Y - AX||_F^2 &= \operatorname{tr}\left((Y-AX)^T(Y-AX)\right) \\ &= \operatorname{tr}\left(X^TA^TAX\right) - \operatorname{tr}\left(X^TA^TY\right) - \operatorname{tr}\left(Y^TAX\right) + \operatorname{tr}\left(Y^TY\right) \end{aligned} \tag{17}$$

Let's calculate the derivative of the traces:

Derivative of the trace of a matrix product AXB wrt X.

Suppose A, B and X are rectangular matrices (different from the ones in the problem) with appropriate dimensions. By definition of the trace operator and of matrix products, $\operatorname{tr}(AXB) = \sum_{ijk} A_{ij}X_{jk}B_{ki}$. The element j', k' of the Jacobian matrix of $\operatorname{tr}(AXB)$:

$$\begin{aligned} \frac{\partial \operatorname{tr}(AXB)}{\partial X_{j'k'}} &= \sum_{ijk} \frac{\partial A_{ij}X_{jk}B_{ki}}{\partial X_{j'k'}}, \text{ the term in the sum is 0 unless } j = j' \text{ and } k = k', \\ &= \sum_i A_{ij'}B_{k'i} = (A^TB^T)_{j'k'} \end{aligned} \tag{18}$$

Thus in matrix notation:

$$\frac{\partial \operatorname{tr}(AXB)}{\partial X} = A^TB^T. \tag{19}$$

In our context, replacing X with A and replacing A, B

$$\frac{\partial \operatorname{tr}\left(Y^TAX\right)}{\partial A} = YX^T \tag{20}$$

Similarly, since:

$$\frac{\partial \operatorname{tr}(AXB)}{\partial X} = \frac{\partial \operatorname{tr}\left(B^TX^TA^T\right)}{\partial X} = A^TB^T \tag{21}$$

replacing X with A, B with X and A^T with Y:

$$\frac{\partial \operatorname{tr}\left(X^TA^TY\right)}{\partial A} = YX^T \tag{22}$$

Derivative of the trace of the matrix product X^TA^TAX wrt A.

We use element notations:

$$\begin{aligned}(AX)_{ij} &= \sum_k A_{ik}X_{kj} \text{ and } (X^TA^T)_{ij} = \sum_k A_{jk}X_{ki} \\ \text{thus },[(X^TA^T)(AX)]_{ij} &= \sum_k (X^TA^T)_{ik}(AX)_{kj} \\ &= \sum_k \left((\sum_l A_{kl}X_{li})(\sum_m A_{km}X_{mj}) \right) \\ &= \sum_{k,l,m} A_{kl}X_{li}A_{km}X_{mj}.\end{aligned} \tag{23}$$

Now we can write the trace:

$$\mathrm{tr}\left(X^TA^TAX\right) = \sum_i \sum_{k,l,m} A_{kl}X_{li}A_{km}X_{mi} = \sum_i \sum_{lm} X_{li}X_{mi} \sum_k A_{km}A_{kl}. \tag{24}$$

Element k', j' of the Jacobian wrt A is:

$$\frac{\partial \mathrm{tr}\left(X^TA^TAX\right)}{\partial A_{k'j'}} = \sum_i \sum_{l,m} X_{li}X_{mi} \sum_k \frac{\partial A_{kl}A_{km}}{\partial A_{k'j'}} \tag{25}$$

The sum on the right hand side is for sure 0 when $k \neq k'$, thus:

$$\frac{\partial \mathrm{tr}\left(X^TA^TAX\right)}{\partial A_{k'j'}} = \sum_i \sum_{l,m} X_{li}X_{mi} \frac{\partial A_{k'l}A_{k'm}}{\partial A_{k'j'}} \tag{26}$$

then we can split the sum on l, m depending on whether l or $m = j'$:

$$\begin{aligned}\frac{\partial \mathrm{tr}\left(X^TA^TAX\right)}{\partial A_{k'j'}} &= \sum_i \sum_{l\neq j'} \left(\sum_{m\neq j'} X_{li}X_{mi}.0 + \sum_{m=j'} X_{li}X_{j'i}\frac{\partial A_{k'l}A_{k'j'}}{\partial A_{k'j'}} \right) \\ &\quad + \sum_{l=j'} \left(\sum_{m\neq j'} X_{j'i}X_{mi}\frac{\partial A_{k'j'}A_{k'm}}{\partial A_{k'j'}} + \sum_{m=j'} X_{j'i}X_{ji}\frac{\partial A^2_{k'j'}}{\partial A_{k'j'}} \right) \\ &= \sum_i \sum_{l\neq j'} X_{li}X_{j'i}A_{k'l} + \sum_{l=j'}(\sum_{m\neq j'} X_{j'i}X_{mi}A_{k'm} + X_{j'i}X_{j'i}2A_{k'j'}) \\ &= \sum_i \left(\sum_l X_{li}X_{j'i}A_{k'l} + \sum_l X_{mi}X_{j'i}A_{k'm} \right)\end{aligned} \tag{27}$$

Here the sum on the l and m index are the same, thus:

$$\begin{aligned}\frac{\partial \mathrm{tr}\left(X^TA^TAX\right)}{\partial A_{k'j'}} &= 2\sum_i \sum_l X_{li}X_{j'i}A_{k'l} \\ &= 2\sum_l A_{k'l} \sum_i X_{li}X_{j'i}\end{aligned} \tag{28}$$

This turns out to be the $k'j'$ index of matrix $2AXX^T$ (you can verify this by writing $2AXX^T$ in index form). Thus in matrix notation:

$$\frac{\partial \mathrm{tr}\left(X^TA^TAX\right)}{\partial A} = 2AXX^T \tag{29}$$

Back to our problem from equation 16:

$$\begin{aligned}\left.\frac{\partial ||Y-AX||_F^2}{\partial A}\right|_{A_n^\star} &= \left.\frac{\partial}{\partial A}\mathrm{tr}\left(X^TA^TAX\right) - \mathrm{tr}\left(X^TA^TY\right) - \mathrm{tr}\left(Y^TAX\right) + \mathrm{tr}\left(Y^TY\right)\right|_{A_n^\star} = 0 \\ &\implies 2A_n^\star XX^T - YX^T - YX^T = 0 \\ &\implies 2(A_n^\star X - Y)X^T = 0 \\ &\implies A_n^\star = YX^T(XX^T)^{-1}\end{aligned} \tag{30}$$

$\blacksquare$

Question 3. The solution above can lead to overfitting, especially when a few data points are provided. Also XX^T may not be invertible. We resort to regularization in these cases. We find $A_n^\star$ such that

$$A_n^\star = \arg\min_A ||Y - AX||_F^2 + \lambda ||A||_F^2, \tag{31}$$

where $\lambda > 0$.

How is $A_n^\star$ calculated in this case ?

Solution: Using previous questions:

$$A_n^\star = YX^T(XX^T + \lambda I)^{-1} \tag{32}$$

$\blacksquare$

Question 4. Implement the solutions to Questions 2 and 3 in Python. Generate data with a linear model and make sure that you are able to recover the linear transformation.

1.4 Kernel based predictions

Similarly to the previous questions, we suppose that we are given n data points for $\mathbf{x}$ and $\mathbf{y}$, in the form of matrices $X \in \mathbb{R}^{d\times n}$ and $Y \in \mathbb{R}^{q\times n}$ respectively. For simplicity we assume q=1.

A kernel based predictor differs from a linear predictor in that it performs linear prediction on a transformed version of the input rather than of the input directly. The transformation is performed by a function (let's call it $\boldsymbol{\phi}$) mapping input vectors to vectors in a space of arbitrary dimension N, i.e. $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^N$. The prediction for a vector $\mathbf{x} \in \mathbb{R}^d$ is written as

$$\hat{\mathbf{y}} = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \tag{33}$$

where $\mathbf{w}$ are parameters of the predictor. In other words, kernel predictors are linear predictors in higher dimensional spaces.

Question 1. Introducing the design matrix.

Derive the MSE solution for $\mathbf{w}$, when constant weighting factors $r_i > 0$ are introduced for each of our samples $\mathbf{x}_i$, i.e. find $\mathbf{w}^\star$ that minimizes:

$$E(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^{n} r_i \left(y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)\right)^2 \tag{34}$$

Question 1a. Write the derivative of E wrt $\mathbf{w}$

Question 1b. Find $\mathbf{w}^*$ that minimizes E as a function of

$$\boldsymbol{\Phi}' = \begin{bmatrix} \sqrt{r_1}\boldsymbol{\phi}(\mathbf{x}_1)^T \\ \vdots \\ \sqrt{r_n}\boldsymbol{\phi}(\mathbf{x}_n)^T \end{bmatrix} \text{ and } Y' = \begin{bmatrix} \sqrt{r_1} y_1 \\ \vdots \\ \sqrt{r_n} y_n \end{bmatrix} \tag{35}$$

Question 1c. How can you interpret the coefficients r_i ?

Solution: The derivative:

$$J = \frac{1}{n} \sum_{i=1}^{n} r_i (y_i - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i)) \boldsymbol{\phi}(\mathbf{x}_i)^T \tag{36}$$

Let us denote $y_i' = \sqrt{r_i} y_i$ and $\boldsymbol{\phi}(\mathbf{x}_i)' = \sqrt{r_i}\boldsymbol{\phi}(\mathbf{x}_i)$. Setting the derivative to 0:

$$0 = \sum_{i=1}^{n} y_i' \boldsymbol{\phi}'(\mathbf{x}_i) - \mathbf{w}^T \left(\sum_{i=1}^{n} \boldsymbol{\phi}'(\mathbf{x}_i)\boldsymbol{\phi}'(\mathbf{x}_i)^T \right) \tag{37}$$

Note that $\boldsymbol{\Phi}'^T\boldsymbol{\Phi}' = \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i)'\boldsymbol{\phi}'(\mathbf{x}_i)^T$ and $\boldsymbol{\Phi}'^T Y' = \sum_{i=1}^{n} y_i'\boldsymbol{\phi}'(\mathbf{x}_i)$ This gives, using the design matrix $\boldsymbol{\Phi}'$: $\mathbf{w}^* = Y'^T\boldsymbol{\Phi}'(\boldsymbol{\Phi}'^T\boldsymbol{\Phi}')^{-1}$.

In imbalanced classification problems, the coefficient r_i can be interpreted as class weights, artificially increasing/decreasing the error made on the under/over-represented class. ■

Question 2. Introducing the Gram matrix.

We now assume that $r_i = 1$ for $i = 1, \dots, n$. Also, we assume that we introduce a regularization parameter $\lambda > 0$ in our MSE.

Question 2a. Write the MSE (similar to equation 34) with a regularization term for $||\mathbf{w}||_2$ and without r_i.

Solution:

$$E(\mathbf{w}) = \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)\right)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{38}$$

■

The design matrix $\boldsymbol{\Phi}$ can be problematic to compute for some choices of $\boldsymbol{\phi}$. Instead let us introduce a way to perform predictions on a new point $\mathbf{x}$, without explicitely writting the design matrix. For this we need another parameter vector :

$$\mathbf{a} \in \mathbb{R}^n, \text{ where } a_i = -\frac{1}{\lambda}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - y_n), \text{ for } i = 1, \dots, n \tag{39}$$

Using this in the expression of the gradient of equation (34), we find that this new parameter vector is related to $\mathbf{w}$ as follows: $\mathbf{w} = \boldsymbol{\Phi}^T\mathbf{a}$.

Question 2b. By introducing the Gram matrix $K = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$, with elements $K_{i,j} = \boldsymbol{\phi}(\mathbf{x}_i)^T\boldsymbol{\phi}(\mathbf{x}_j)$ show that the prediction for a vector $\mathbf{x} \in \mathbb{R}^d$ can be obtained as (Eq (6.9) in Bishop):

$$\hat{\mathbf{y}} = Y\left(K + \lambda I_n\right)^{-1} \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}) \end{bmatrix} \tag{40}$$

you can assume $q = 1$ for simplicity.

Solution: We re-write equation 34:

$$\begin{aligned}
E(\mathbf{w}) &= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i) - y_i)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \\
&= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i))^2 - \sum_{i=1}^{n}\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)y_i + \frac{1}{2}\sum_{i=1}^{n}y_i^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \\
&= \frac{1}{2}\sum_{i=1}^{n}(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)).(\boldsymbol{\phi}(\mathbf{x}_i)^T\mathbf{w}) - \sum_{i=1}^{n}\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)y_i + \frac{1}{2}\sum_{i=1}^{n}y_i^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \\
&= \frac{1}{2}\mathbf{w}^T\left(\sum_{i=1}^{n}\boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_i)^T\right)\mathbf{w} - \sum_{i=1}^{n}\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)y_i + \frac{1}{2}\sum_{i=1}^{n}y_i^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \\
&= \frac{1}{2}\mathbf{a}^T\boldsymbol{\Phi}\left(\sum_{i=1}^{n}\boldsymbol{\phi}(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_i)^T\right)\boldsymbol{\Phi}^T\mathbf{a} - \mathbf{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^TY^T + \frac{1}{2}Y^TY + \frac{\lambda}{2}\mathbf{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{a} \\
&= \frac{1}{2}\mathbf{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{a} - \mathbf{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^TY^T + \frac{1}{2}Y^TY + \frac{\lambda}{2}\mathbf{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{a}
\end{aligned} \tag{41}$$

The Gram matrix is introduced as $K = \mathbf{\Phi}\mathbf{\Phi}^T$ which leads to

$$E(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T KK\mathbf{a} - \mathbf{a}^T KY^T + \frac{1}{2}Y^T Y + +\frac{\lambda}{2}\mathbf{a}^T K\mathbf{a} \tag{42}$$

Deriving wrt $\mathbf{a}$ and setting the gradient to 0:

$$\mathbf{a}^\star = (K + \lambda I_n)^{-1} Y. \tag{43}$$

This leads to the equation for the prediction of a target from a new input $\mathbf{x}$:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}^T \mathbf{\Phi}\boldsymbol{\phi}(\mathbf{x}) = Y^T(K + \lambda I_n)^{-1} \begin{bmatrix} \boldsymbol{\phi}^T(\mathbf{x}_1) \\ \vdots \\ \boldsymbol{\phi}^T(\mathbf{x}_n) \end{bmatrix} \boldsymbol{\phi}(\mathbf{x}) \\ \hat{\mathbf{y}} &= Y^T(K + \lambda I_n)^{-1} \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}) \end{bmatrix} \end{aligned} \tag{44}$$

■

Question 3. Implement a function in Python that calculates the Gram matrix associated with a linear kernel. Your function should take as argument two sets of vectors in $\mathbb{R}^d$ in the form of two matrices, e.g. $X_1 \in \mathbb{R}^{d\times n}$ and $X_2 \in \mathbb{R}^{d\times m}$, and return the Gram matrix $K \in \mathbb{R}^{n\times m}$.

Question 4. Similarly, implement a function in Python that calculates the Gram matrix associated with a RBF kernel.

Question 5. Implement (40). Your function should take a matrix with input points columnwise and return a matrix with the predicted vectors columnwise.

Question 6. Compare the performances of a kernel predictor with a linear kernel and with a RBF kernel. You can use a toy dataset for this, e.g.:

```
from sklearn.datasets import make_circles;
X,Y  = make_circles(n_samples=1_000, factor=0.3, noise=0.05, random_state=0);
```