EECS 553: Machine Learning (ECE) **University of Michigan**

Kernel Ridge Regression

Winter 2023 **Clayton Scott**

In these notes, we will study a very useful, general-purpose method for nonlinear regression known as *kernel ridge regression* (KRR). KRR is the result of kernelizing ridge regression. Ridge regression is not usually expressed in a form that is obviously kernelizable, and so we must do some work to put it in that form.

1 KRR without Offset

In the interest of simplicity, we begin by kernelizing ridge regression *without* offset. In other words, the bias/offset parameter b is assumed to be zero, and the regression model is $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$. This model is not simply for pedagogical purposes. If the feature map Φ associated to a kernel contains a constant element (such as an inhomogeneous polynomial kernel), then the offset is redundant. And even if this is not the case, if the response variables have been centered (i.e., made to have zero mean), then kernels like the Gaussian kernel (whose feature map does *not* contain a constant element) still lead to effective regression estimates without an offset.

Ridge regression without offset minimizes

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^T\boldsymbol{x}_i)^2 + \lambda\left\|\boldsymbol{w}\right\|^2 &= \frac{1}{n}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2 \\ &\propto \boldsymbol{w}^T(\boldsymbol{X}^T\boldsymbol{X} + n\lambda\boldsymbol{I})\boldsymbol{w} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T\boldsymbol{y},\end{aligned}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \qquad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix}.$$

The solution is

$$\widehat{\boldsymbol{w}} = (\boldsymbol{X}^T\boldsymbol{X} + n\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

and the regression function estimate is:

$$\widehat{f}(\boldsymbol{x}) = \widehat{\boldsymbol{w}}^T\boldsymbol{x} = \boldsymbol{y}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X} + n\lambda\boldsymbol{I})^{-1}\boldsymbol{x}. \tag{1}$$

To kernelize this method, we must show that $\widehat{f}(\boldsymbol{x})$ depends on $\boldsymbol{x}, \boldsymbol{x}_1, ..., \boldsymbol{x}_n$ only in terms of inner products of the form $\langle \boldsymbol{u}, \boldsymbol{v}\rangle$ where $\boldsymbol{u}, \boldsymbol{v} \in \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{x}\}$. This is not obvious from (1). The matrix $\boldsymbol{X}^T\boldsymbol{X}$ is *not* the Gram matrix.

We will use the *matrix inversion lemma*, which states that

$$(\boldsymbol{P} + \mathbf{QRS})^{-1} = \boldsymbol{P}^{-1} - \boldsymbol{P}^{-1}\boldsymbol{Q}(\boldsymbol{R}^{-1} + \mathbf{SP}^{-1}\boldsymbol{Q})^{-1}\mathbf{SP}^{-1}.$$

Substituting $\mu = n\lambda$ for brevity, we apply this identity with $\boldsymbol{P} = \mu\boldsymbol{I}$, $\boldsymbol{Q} = \boldsymbol{X}^T$, $\boldsymbol{R} = \boldsymbol{I}$, $\boldsymbol{S} = \boldsymbol{X}$, which yields

$$\begin{aligned}(\mu\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{X})^{-1} &= \frac{1}{\mu}\boldsymbol{I} - \frac{1}{\mu}\boldsymbol{I}\boldsymbol{X}^T(\boldsymbol{I} + \frac{1}{\mu}\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\frac{1}{\mu}\boldsymbol{I} \\ &= \frac{1}{\mu}[\boldsymbol{I} - \boldsymbol{X}^T(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}].\end{aligned}$$

This shows

$$\widehat{\boldsymbol{w}}^T = \frac{1}{\mu}\boldsymbol{y}^T\boldsymbol{X}[\boldsymbol{I} - \boldsymbol{X}^T(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}].$$

This expression can be simplified as follows:

$$\begin{aligned}
\widehat{\boldsymbol{w}}^T &= \frac{1}{\mu}\boldsymbol{y}^T\boldsymbol{X}[\boldsymbol{I} - \boldsymbol{X}^T(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}] \\
&= \frac{1}{\mu}\boldsymbol{y}^T[\boldsymbol{X} - \boldsymbol{X}\boldsymbol{X}^T(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}] \\
&= \frac{1}{\mu}\boldsymbol{y}^T[\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^T(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}]\boldsymbol{X} \\
&= \frac{1}{\mu}\boldsymbol{y}^T[(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1} - \boldsymbol{X}\boldsymbol{X}^T(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}]\boldsymbol{X} \\
&= \frac{1}{\mu}\boldsymbol{y}^T[(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T - \boldsymbol{X}\boldsymbol{X}^T)(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}]\boldsymbol{X} \\
&= \frac{1}{\mu}\boldsymbol{y}^T[\mu\boldsymbol{I}(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}]\boldsymbol{X} \\
&= \boldsymbol{y}^T(\mu\boldsymbol{I} + \boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}
\end{aligned}$$

Now the Gram matrix $\boldsymbol{G} = \boldsymbol{X}\boldsymbol{X}^T$ appears, although the method is still not kernelized because of the matrix $\boldsymbol{X}$. The training vectors that constitute the rows of this matrix are not inside of an inner product. Fortunately, this issue is resolved when we take the inner product of $\widehat{\boldsymbol{w}}$ with a test instance $\boldsymbol{x}$. Thus, introduce the notation

$$\boldsymbol{G} := \begin{bmatrix} \langle \boldsymbol{x}_1, \boldsymbol{x}_1 \rangle & \cdots & \langle \boldsymbol{x}_1, \boldsymbol{x}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \boldsymbol{x}_n, \boldsymbol{x}_1 \rangle & \cdots & \langle \boldsymbol{x}_n, \boldsymbol{x}_n \rangle \end{bmatrix} \qquad \boldsymbol{g}(\boldsymbol{x}) := \begin{bmatrix} \langle \boldsymbol{x}_1, \boldsymbol{x} \rangle \\ \vdots \\ \langle \boldsymbol{x}_n, \boldsymbol{x} \rangle \end{bmatrix}.$$

Then we have

$$\begin{aligned}
\widehat{f}(\boldsymbol{x}) &= \widehat{\boldsymbol{w}}^T\boldsymbol{x} \\
&= \boldsymbol{y}^T(\boldsymbol{X}\boldsymbol{X}^T + n\lambda\boldsymbol{I})^{-1}\boldsymbol{X}\boldsymbol{x} \\
&= \boldsymbol{y}^T(\boldsymbol{G} + n\lambda\boldsymbol{I})^{-1}\boldsymbol{g}(\boldsymbol{x}).
\end{aligned}$$

This shows that KRR w/o offset is kernelizable.

To kernelize the method, we simply select a kernel k and replace $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ with $k(\boldsymbol{u}, \boldsymbol{v})$ in the definitions of $\boldsymbol{G}$ and $\boldsymbol{g}(\boldsymbol{x})$. After making this substitution, $\boldsymbol{G}$ and $\boldsymbol{g}(\boldsymbol{x})$ are replaced by

$$\boldsymbol{K} := \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{bmatrix}, \qquad \boldsymbol{k}(\boldsymbol{x}) := \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}) \\ \vdots \\ k(\boldsymbol{x}_n, \boldsymbol{x}) \end{bmatrix}.$$

$\boldsymbol{K}$ is called the *kernel matrix*. Now, the final form of the KRR (w/o offset) predictor is

$$\widehat{f}(\boldsymbol{x}) = \boldsymbol{y}^T(\boldsymbol{K} + n\lambda\boldsymbol{I})^{-1}\boldsymbol{k}(\boldsymbol{x}).$$

Recall that the entire point of this exercise is to obtain a nonlinear regression method. If k is such that $k(\boldsymbol{u}, \boldsymbol{v}) = \langle \Phi(\boldsymbol{u}), \Phi(\boldsymbol{v}) \rangle$, then the above kernel method is equivalent to first applying the feature map Φ to all feature vectors, and then applying ridge regression w/o offset in the new feature space. Because we have kernelized the method, we do not have to actually compute or work with Φ, all calculations are in terms of the kernel.

The computational complexity of KRR without offset is $O(n^3)$ which comes from having to invert an $n \times n$ matrix. As with regular ridge regression, this can be accelerated using gradient descent and related methods.

2 Kernel Ridge Regression with Offset

The derivation of kernel ridge regression *with* offset is similar KRR without offset, but with one important additional concept. We have previously seen that the solution to ridge regression with offset is

$$\widehat{f}(\boldsymbol{x}) = \widehat{\boldsymbol{w}}^T \boldsymbol{x} + \widehat{b}$$

where

$$\begin{aligned}\widehat{\boldsymbol{w}} &= (\tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}} + n\lambda \boldsymbol{I})^{-1} \tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{y}} \\ \widehat{b} &= \bar{y} - \widehat{\boldsymbol{w}}^T \bar{\boldsymbol{x}}\end{aligned}$$

and

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_n \end{bmatrix}, \quad \tilde{y}_i = y_i - \bar{y}, \qquad \tilde{\boldsymbol{X}} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1^T \\ \vdots \\ \tilde{\boldsymbol{x}}_n^T \end{bmatrix}, \quad \tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}.$$

Here $\bar{\boldsymbol{x}} = \frac{1}{n}\sum_i \boldsymbol{x}_i$ and $\bar{y} = \frac{1}{n}\sum_i y_i$. The regression function estimate is then

$$\widehat{f}(\boldsymbol{x}) = \widehat{\boldsymbol{w}}^T \boldsymbol{x} + \widehat{b} = \bar{y} + \widehat{\boldsymbol{w}}^T (\boldsymbol{x} - \bar{\boldsymbol{x}}). \tag{2}$$

To kernelize this function, we can follow the exact same steps as for KRR without offset to arrive at

$$\widehat{\boldsymbol{w}}^T = \tilde{\boldsymbol{y}}^T (\tilde{\boldsymbol{G}} + n\lambda \boldsymbol{I})^{-1} \tilde{\boldsymbol{X}}$$

and

$$\widehat{f}(\boldsymbol{x}) = \bar{y} + \tilde{\boldsymbol{y}}^T \left(\tilde{\boldsymbol{G}} + n\lambda \boldsymbol{I}\right)^{-1} \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}), \tag{3}$$

where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}}$ and

$$\tilde{\boldsymbol{G}} := \begin{bmatrix} \langle \tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_1 \rangle & \cdots & \langle \tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \tilde{\boldsymbol{x}}_n, \tilde{\boldsymbol{x}}_1 \rangle & \cdots & \langle \tilde{\boldsymbol{x}}_n, \tilde{\boldsymbol{x}}_n \rangle \end{bmatrix}, \qquad \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}) := \begin{bmatrix} \langle \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_1 \rangle \\ \vdots \\ \langle \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}_n \rangle \end{bmatrix}.$$

Basically KRR w/ offset is like KRR w/o offset applied to the mean-centered feature space. In additon, $\bar{y}$ is added to the predicted output.

Expanding the entries of $\tilde{\boldsymbol{G}}$ and $\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}})$ we have

$$\begin{aligned}\langle \tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j \rangle &= \langle \boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{x}_j - \bar{\boldsymbol{x}} \rangle \\ &= \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle - \frac{1}{n}\sum_{r=1}^{n} \langle \boldsymbol{x}_i, \boldsymbol{x}_r \rangle - \frac{1}{n}\sum_{s=1}^{n} \langle \boldsymbol{x}_s, \boldsymbol{x}_j \rangle + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} \langle \boldsymbol{x}_r, \boldsymbol{x}_s \rangle\end{aligned}$$

and

$$\begin{aligned}\langle \tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}} \rangle &= \langle \boldsymbol{x}_i - \bar{\boldsymbol{x}}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle \\ &= \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle - \frac{1}{n}\sum_{r} \langle \boldsymbol{x}_i, \boldsymbol{x}_r \rangle - \frac{1}{n}\sum_{s} \langle \boldsymbol{x}, \boldsymbol{x}_s \rangle + \frac{1}{n^2}\sum_{r}\sum_{s} \langle \boldsymbol{x}_r, \boldsymbol{x}_s \rangle.\end{aligned}$$

Therefore, to kernelize ridge regression with offset, we select a kernel k and replace $\tilde{\boldsymbol{G}}$ with $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{g}}(\boldsymbol{x})$ with $\tilde{\boldsymbol{k}}(\boldsymbol{x})$, where the (i,j) entry of $\tilde{\boldsymbol{K}}$ is

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{n}\sum_{r=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_r) - \frac{1}{n}\sum_{s=1}^{n} k(\boldsymbol{x}_s, \boldsymbol{x}_j) + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} k(\boldsymbol{x}_r, \boldsymbol{x}_s),$$

and the ith entry of $\tilde{\boldsymbol{k}}(\tilde{\boldsymbol{x}})$ is

$$k(\boldsymbol{x}_i, \boldsymbol{x}) - \frac{1}{n}\sum_r k(\boldsymbol{x}_i, \boldsymbol{x}_r) - \frac{1}{n}\sum_s k(\boldsymbol{x}, \boldsymbol{x}_s) + \frac{1}{n^2}\sum_r\sum_s k(\boldsymbol{x}_r, \boldsymbol{x}_s).$$

Thus, the final KRR (w/ offset) predictor is

$$\widehat{f}(\boldsymbol{x}) = \bar{y} + \tilde{\boldsymbol{y}}^T(\tilde{\boldsymbol{K}} + n\lambda \boldsymbol{I})^{-1}\tilde{\boldsymbol{k}}(\boldsymbol{x}).$$

As an exercise you are asked to show that $\tilde{\boldsymbol{K}}$ and $\tilde{\boldsymbol{k}}(\tilde{\boldsymbol{x}})$ can be calculated, using a simple formula, in terms $\boldsymbol{K}$ and $\boldsymbol{k}(\boldsymbol{x})$.

Remark 1. From Eqn. (3), it is *very tempting* to attempt to kernelize this method by replacing dot products $\langle \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}' \rangle$ with $k(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}')$, where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \bar{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{x}} - \bar{\boldsymbol{x}}$. However, this is incorrect. To see why, let us introduce the notation

$$\tilde{\Phi}(\boldsymbol{x}) := \Phi(\boldsymbol{x}) - \frac{1}{n}\sum_i \Phi(\boldsymbol{x}_i).$$

The formula for $\widehat{\boldsymbol{w}}$ requires us to subtract the mean off of all feature vectors. When we kernelize a method, we are effectively replacing all features vectors $\boldsymbol{x}$ with $\Phi(\boldsymbol{x})$. Therefore, to calculate the formula for $\widehat{\boldsymbol{w}}$ based on the transformed data, we must subtract the mean off of all feature vectors *after transforming to the new feature space*. In other words, $\tilde{\boldsymbol{x}}$ in the original feature space gets mapped to $\tilde{\Phi}(\boldsymbol{x})$ in the new feature space, and $\langle \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}' \rangle$ is replaced by $\langle \tilde{\Phi}(\boldsymbol{x}), \tilde{\Phi}(\boldsymbol{x}') \rangle$. This agrees with how the algorithm was kernelized above, which you are asked to show as an exercise.

In contrast, the formula $k(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}')$ amounts to computing $\langle \Phi(\tilde{\boldsymbol{x}}), \Phi(\tilde{\boldsymbol{x}}') \rangle$. Because Φ is nonlinear except in trivial cases, we generally have $\Phi(\tilde{\boldsymbol{x}}) \neq \tilde{\Phi}(\boldsymbol{x})$, and so this approach does not kernelize the method.

We refer to $\tilde{\Phi}$ as the *centered feature map*, $\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') := \langle \tilde{\Phi}(\boldsymbol{x}), \tilde{\Phi}(\boldsymbol{x}') \rangle$ as the *centered kernel*, and $\tilde{\boldsymbol{K}}$ as the *centered kernel matrix* associated to the training data set.

3 Gaussian Kernel

To gain some insight into kernel ridge regression, suppose the kernel is a Gaussian kernel and assume there is no offset. Then the nonlinear function estimate may be expressed

$$\begin{aligned}\widehat{f}(\boldsymbol{x}) &= \boldsymbol{\alpha}^T\boldsymbol{k}(\boldsymbol{x}) \\ &= \sum_i \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i).\end{aligned}$$

where $\boldsymbol{\alpha} = (n\lambda\boldsymbol{I} + \boldsymbol{K})^{-1}\boldsymbol{y}$. Viewing $\boldsymbol{x}$ as a variable and $\boldsymbol{x}_i$ as fixed, we can view $k(\boldsymbol{x}, \boldsymbol{x}_i)$ as a bell-shaped function whose peak is at $\boldsymbol{x}_i$. Therefore, the KRR estimate is a linear combination of such functions. The class of all such functions (where $\boldsymbol{\alpha}$ is allowed to vary) is a large class of nonlinear functions. See Figure 1.

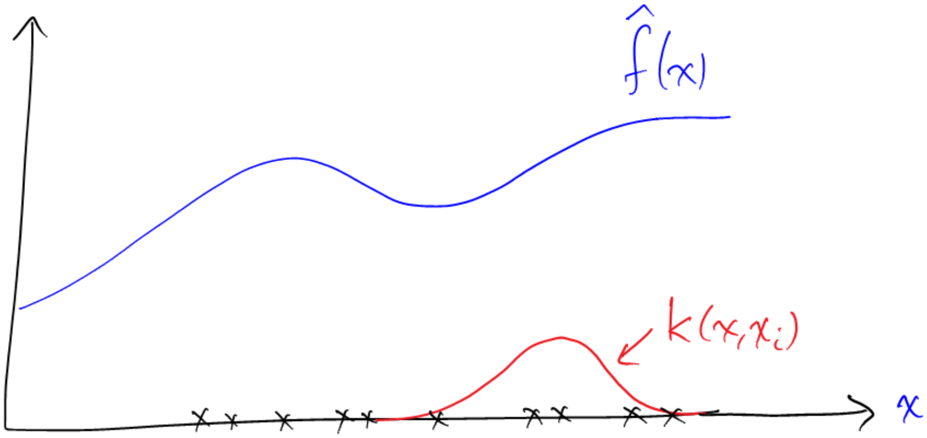
Exercises

1. (★★) What is the difference between
   - KRR without offset using the kernel $k(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}^T\boldsymbol{v} + 1)^2$, and
   - KRR with offset using the kernel $k(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}^T\boldsymbol{v})^2$?

2. (★★) In KRR with offset, give a formula for the offset b using the kernel.

3. (★) Show that in KRR w/ offset, the entries of $\tilde{\boldsymbol{K}}$ can be expressed $\langle \tilde{\Phi}(\boldsymbol{x}_i), \tilde{\Phi}(\boldsymbol{x}_j) \rangle$, where $\tilde{\Phi}(\boldsymbol{x}) := \Phi(\boldsymbol{x}) - \frac{1}{n}\sum_{\ell=1}^{n}\Phi(\boldsymbol{x}_\ell)$.

Figure 1: KRR estimate with Gaussian kernel. The final estimate is a weighted sum of Gaussians centered at the training points.



4. (★★) This problem investigates concise formulas for working with the centered kernel matrix.

(a) Show that

$$\tilde{\boldsymbol{K}} = \boldsymbol{K} - \boldsymbol{K}\boldsymbol{O} - \boldsymbol{O}\boldsymbol{K} + \boldsymbol{O}\boldsymbol{K}\boldsymbol{O}, \tag{4}$$

where $\boldsymbol{O}$ is a square matrix with all entries equal to $1/n$. Note that $\boldsymbol{O}$ may be expressed $\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$, where $\mathbf{1}_n \in \mathbb{R}^n$ is the column vector of all ones.

(b) Now consider a test data set $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_m$, and let $\boldsymbol{K}'$ be the $n \times m$ train-test matrix with entries $k(\boldsymbol{x}_i, \boldsymbol{x}'_j)$, and let $\tilde{\boldsymbol{K}}'$ be the $n \times m$ centered train-test matrix, whose entries are $k(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}'_j)$. Determine a formula analogous to (4) for relating $\tilde{\boldsymbol{K}}'$ to $\boldsymbol{K}'$. Note that $\tilde{\boldsymbol{x}}'_j = \boldsymbol{x}'_j - \bar{\boldsymbol{x}}$ where $\bar{\boldsymbol{x}}$ is still the sample mean of the *training data.*

(c) Use the previous result to determine a formula for computing the predicted outputs (according to KRR with offset) on all the test points. Your formula should yield a column vector of length m.