IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 December 2023, accepted 11 January 2024, date of publication 22 January 2024, date of current version 6 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3357097

THEORY

A New Formula for Faster Computation of the K-Fold Cross-Validation and Good Regularisation Parameter Values in Ridge Regression

KRISTIAN HOVDE LILAND, JOAKIM SKOGHOLT, AND ULF GEIR INDAHL
Faculty of Science and Technology, Norwegian University of Life Sciences, 1432 Ås, Norway

Corresponding author: Kristian Hovde Liland (kristian.liland@nmbu.no)

This work was supported by the Research Council of Norway under Project 239070.

ABSTRACT In the present paper, we prove a new theorem, resulting in an update formula for linear regression model residuals calculating the exact k-fold cross-validation residuals for any choice of cross-validation strategy without model refitting. The required matrix inversions are limited by the cross-validation segment sizes and can be executed with high efficiency in parallel. The well-known formula for leave-one-out cross-validation follows as a special case of the theorem. In situations where the cross-validation segments consist of small groups of repeated measurements, we suggest a heuristic strategy for fast serial approximations of the cross-validated residuals and associated Predicted Residual Sum of Squares (*PRESS*) statistic. We also suggest strategies for efficient estimation of the minimum *PRESS* value and full *PRESS* function over a selected interval of regularisation values. The computational effectiveness of the parameter selection for Ridge- and Tikhonov regression modelling resulting from our theoretical findings and heuristic arguments is demonstrated in several applications with real and highly multivariate datasets.

INDEX TERMS Cross-validation, GCV, PRESS statistic, ridge regression, SVD, Tikhonov regularisation.

I. INTRODUCTION

Model-/parameter selection in statistical modelling is frequently justified from the maximum likelihood (ML) principle in combination with some measure of model quality (such as the Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), Mallows C_p, the *PRESS* statistic, etc.) that estimates the expected predictive performance for some candidate model(s) [1].

According to Hjorth [2] the application of cross-validation measures as a methodology for model-/parameter selection in statistical applications was introduced by Stone [3]. Stone's ideas motivated the invention of the generalised cross-validation (*GCV*) method by Golub et al. [4] which is a computationally efficient approximation to the leave-one-out cross-validation (LooCV) method. It is invariant under orthogonal transformations and is considered to be a computationally efficient method for choosing appropriate regularisation parameter values in ridge regression (RR) modelling. Cross-validation is still an active area of research, see, e.g., [5], [6], and [7] for some recent works regarding prediction estimates for cross-validation, and [8] for an analysis of the stability of generalisation bounds for leave-one-out cross-validation. The focus of the present work is the selection of an appropriate regularisation parameter value, primarily by grid search but numerical optimisation is also discussed.

The RR method was introduced to the statistics community by Hoerl and Kennard [9], and is perhaps the most important special case in the Tikhonov regularisation [10] (TR) framework of linear regression methods. The TR ideas were originally introduced to the community

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

of numerical mathematics for solving linear discrete ill-posed problems in the context of inverse modelling. A good elementary introduction to the field is given in Hansen [11].

The fast and exact calculations of the LooCV based *Predicted Residual Sum of Squares* (*PRESS*) statistic for the ordinary least squares (OLS) regression have been demonstrated by Allen [12], [13]. The purpose of the present paper is to demonstrate that such calculations are also available for the regularisation parameter selection problem of TR/RR at essentially no additional computational cost. In the present paper, we demonstrate this as follows:

i) From the Sherman–Morrison–Woodbury updating formula for matrix inversion, see Householder [14], we prove a new theorem that gives the general formula for calculating the segmented cross-validation (SegCV) residuals of linear least squares regression modelling. The formula for calculating the LooCV residuals in Allen's *PRESS* statistic [12], [13] follows as a corollary of this result.

ii) We demonstrate how to obtain simple and fast LooCV calculations utilising the compact singular value decomposition (SVD) of a data matrix to quickly obtain *PRESS* values associated with any choice of the regularisation parameter for a TR-problem. In particular, this enables fast graphing of the *PRESS*-values as a function of the regularisation parameter at any desired level of detail.

iii) For situations where some segmented cross-validation approach is required for obtaining the relevant *PRESS*-statistic values in the regularisation parameter selection, one may experience that even the segmented cross-validation formula from our theorem becomes computationally slow. To handle such situations, we propose an approximation of the segmented (K-fold) cross-validation strategy by invoking the computationally inexpensive LooCV strategy after conducting an appropriate orthogonal transformation of the data matrix. The particular orthogonal transformation is constructed from the left singular vectors of the K local SVDs associated with each of the K distinct cross-validation segments.

We demonstrate that the latter alternative provides practically useful approximations of the *PRESS*-statistic at substantial computational savings – in particular for large datasets with many cross-validation segments (large K) containing either identical or highly related measurement values.

II. MATHEMATICAL PRELIMINARIES

If not otherwise stated we assume that $\mathbf{X}$ is a centred ($n \times p$) data matrix ($\mathbf{X}'$ denotes the transpose of $\mathbf{X}$) and that the corresponding ($n \times 1$) vector $\mathbf{y}$ of responses is also centred. We define the scalar $\bar{y}$ and row vector $\bar{\mathbf{x}}$ as the (column) averages of $\mathbf{y}$ and $\mathbf{X}$ obtained before centring, respectively.

A. MODEL ESTIMATION IN ORDINARY LEAST SQUARES AND RIDGE REGRESSION

In ordinary least squares (OLS) regression [1] one minimises the *residual sum of squares*

$$RSS(\mathbf{b}) = \|\mathbf{Xb} - \mathbf{y}\|^2, \tag{1}$$

to identify the least squares solution(s) of (1) with respect to the regression coefficients $\mathbf{b}$. A least squares solution $\mathbf{b}_{OLS}$ of (1) corresponds to an exact solution of the associated *normal equations*

$$\mathbf{X}'\mathbf{Xb} = \mathbf{X}'\mathbf{y}, \tag{2}$$

where $\mathbf{b}_{OLS}$ is unique when $\mathbf{X}'\mathbf{X}$ is non-singular. For later predictions of uncentred data, the associated vector of fitted values is given by

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}_{OLS} + b_0, \tag{3}$$

where the constant term (intercept) $b_0 = \bar{y} - \bar{\mathbf{x}}\mathbf{b}_{OLS}$.

For centred vectors/matrices, $\mathbf{y}$ and $\mathbf{X}$, this equation becomes $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}_{OLS} = \mathbf{Hy}$. Here, the projection matrix, $\mathbf{H}$, (a.k.a. the hat matrix) is defined as

$$\mathbf{H} \overset{\text{def}}{=} \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \mathbf{TT}', \tag{4}$$

where $\mathbf{T}$ can be chosen as any orthogonal ($n \times r$)-matrix spanning the column space of the centred $\mathbf{X}$-data.

For various reasons a minimiser $\mathbf{b}_{OLS}$ of $RSS(\mathbf{b})$ in equation (1) is not always the most attractive choice from a predictive point of view [1], [11], [15]. For instance $\mathbf{X}'\mathbf{X}$ may be singular or poorly conditioned, the solution of (2) is not unique or inappropriate etc. An alternative and quite useful solution was independently recognised by Tikhonov [10], Phillips [16], and Hoerl and Kennard [9]. Instead of directly minimising $RSS(\mathbf{b})$, their alternative proposal was to minimise the weighted bi-objective least squares problem

$$RSS_\lambda(\mathbf{b}) = \|\mathbf{Xb} - \mathbf{y}\|^2 + \lambda\|\mathbf{Ib} - \mathbf{0}\|^2 = \|\mathbf{Xb} - \mathbf{y}\|^2 + \lambda\|\mathbf{b}\|^2, \tag{5}$$

where the scalar $\lambda > 0$ is a fixed *regularisation parameter* (of appropriate magnitude), the matrix $\mathbf{I}$ is the ($p \times p$) identity matrix and $\mathbf{0}$ is a ($p \times 1$) vector of zeros. This formulation explicitly represents a penalisation with respect to the Euclidean (L_2) norm $\|\mathbf{b}\|$ of the regression coefficients. The identity matrix $\mathbf{I}$ can also be replaced by an alternative regularisation matrix $\mathbf{L}$ as described in Appendix C. For a fixed λ, the unique minimiser of (5) is given by $\mathbf{b}_\lambda$ of equation (8) below. The rightmost part of Equation (5) is sometimes referred to as a TR-problem in *standard form* [11].

The minimisation of equation (5) with respect to $\mathbf{b}$ is equivalent to solving the OLS problem associated with the augmented data matrix and response vector:

$$\mathbf{X}_\lambda = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}, \quad \mathbf{y}_0 = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}. \tag{6}$$

Note that linear independence of the $\mathbf{X}_\lambda$-columns trivially follows from linear independence of the $\mathbf{I}$-columns. The matrix product $\mathbf{X}_\lambda'\mathbf{X}_\lambda$ in the associated normal equations

$$\mathbf{X}_\lambda'\mathbf{X}_\lambda\mathbf{b} = \mathbf{X}_\lambda'\mathbf{y}_0 \tag{7}$$

is therefore non-singular, and the corresponding least squares solution

$$\mathbf{b}_\lambda = (\mathbf{X}_\lambda'\mathbf{X}_\lambda)^{-1}\mathbf{X}_\lambda'\mathbf{y}_0 \tag{8}$$

of the augmented problem (6) becomes unique. Straight forward algebraic simplifications of (7) result in the the familiar normal equations associated with the RR-problem

$$(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})\mathbf{b} = \mathbf{X}'\mathbf{y}, \tag{9}$$

and the solution in (8) simplifies to

$$\mathbf{b}_\lambda = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y}. \tag{10}$$

For subsequent applications of the λ-regularised model to uncentred $\mathbf{X}$-data, the appropriate constant term in the resulting regression model is

$$b_{0,\lambda} = \bar{y} - \bar{\mathbf{x}}\mathbf{b}_\lambda, \tag{11}$$

and the associated vector of fitted values $\hat{\mathbf{y}}_\lambda$ is given by

$$\hat{\mathbf{y}}_\lambda = \mathbf{X}\mathbf{b}_\lambda + b_{0,\lambda}. \tag{12}$$

The full SVD of $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}'$ yields $\mathbf{V}\mathbf{V}' = \mathbf{I}_p$ and $\mathbf{X}'\mathbf{X} = \mathbf{V}\mathbf{S}'\mathbf{S}\mathbf{V}'$. Assuming that $\mathbf{X}$ has full rank it is shown in Appendix A that the regression coefficients are given by $\mathbf{b}_\lambda = \mathbf{V}_r\mathbf{c}_\lambda$, where $\mathbf{V}_r$ are the right singular vectors of the compact SVD and the coordinate vector $\mathbf{c}_\lambda$ has the scalar entries

$$c_{\lambda,j} = \frac{\mathbf{u}_j'\mathbf{y}}{s_j + \lambda/s_j}, \text{ for } 1 \leq j \leq r. \tag{13}$$

Compared to the relatively large computational costs associated with calculating the (compact) SVD of $\mathbf{X}$, calculation of the regression coefficient candidates (even for a large number of different λ-values) only requires computing the vectors $\mathbf{c}_\lambda$ according to Equation (37) and the matrix-vector multiplications $\mathbf{b}_\lambda = \mathbf{V}_r\mathbf{c}_\lambda$ as derived in Equation (36).

For the regularised multivariate regression with several (q) responses, $\mathbf{Y} \in \mathbb{R}^{n\times q}$, the associated matrix of regression coefficients is

$$[\mathbf{b}_{1,\lambda}\ \ldots\ \mathbf{b}_{q,\lambda}] = \mathbf{V}_r(\mathbf{S}_r + \lambda\mathbf{S}_r^{-1})^{-1}\mathbf{U}_r'\mathbf{Y} = \mathbf{V}_r\mathbf{C}_\lambda, \tag{14}$$

where $\mathbf{C}_\lambda = (\mathbf{S}_r + \lambda\mathbf{S}_r^{-1})^{-1}\mathbf{U}_r'\mathbf{Y}$ is the obvious multivariate generalisation of the vector $\mathbf{c}_\lambda$ introduced above.

B. OBTAINING CROSS-VALIDATION SEGMENTS BY PROJECTION MATRIX CORRECTION

When the columns of the data matrix $\mathbf{X}$ are linearly independent, the associated OLS-solution $\mathbf{b}_{OLS}$ of the normal equations (2) is unique, and cross-validation residuals can be derived from the Sherman–Morrison–Woodbury formula for updating matrix inverses [14]. From Theorem B in the Appendix, we obtain the general segmented CV (SegCV) residuals

$$\mathbf{r}_{(\{k\})} = [\mathbf{I}_{n_k} - \mathbf{H}_{\{k\}}]^{-1}\mathbf{r}_{\{k\}}, \tag{15}$$

where $\{k\}$ refers to the samples of the k-th CV segment, $\mathbf{r}_{(\{k\})}$ refers to the vector of predicted residuals when the segment samples are not included in the modelling, n_k is the number of samples in the segment and, $\mathbf{H}_{\{k\}}$ is the sub-matrix of the projection matrix $\mathbf{H}$ (defined in Equation (4) above) associated with the samples of the k-th CV segment. This means that updating residuals for a given segment entails the inversion of a matrix involving the entries of $\mathbf{H}$ corresponding to all pairs of sample indices of the k-th CV segment. The computational cost of the inversions obviously depends on the number of segments and the number of samples belonging to each segment.

Allen [12], [13] suggested the *PRESS* (Prediction Sum-Of-Squares) statistic

$$PRESS = \sum_{i=1}^{n}(y_i - \hat{y}_{i,(i)})^2 = \sum_{i=1}^{n}\mathbf{r}_{(i)}'\mathbf{r}_{(i)}. \tag{16}$$

where $\hat{y}_{i,(i)}$ denotes the OLS prediction of the i-th sample when the sample has been deleted from the regression estimation, and $\mathbf{r}_{(i)}$ is the corresponding residual. With $\hat{y}_{i,(\{k\})}$ denoting the predictions of the i-th sample after deleting the corresponding k-th CV segment samples from the regression problem in (1), the SegCV equivalent of the *PRESS*-statistic becomes:

$$PRESS = \sum_{i=1}^{n}(y_i - \hat{y}_{i,(\{k\})})^2 = \sum_{k=1}^{K}\mathbf{r}_{(\{k\})}'\mathbf{r}_{(\{k\})}$$
$$= \sum_{k=1}^{K}\sum_{i=1}^{n_k} r_{i,\{k\}}^2. \tag{17}$$

Here $r_{i,\{k\}}$ are the elements of the residual vectors defined in Equation 15.

1) THE LEAVE-ONE-OUT CROSS-VALIDATION

Corollary B of Theorem B covers the special case of LooCV where Equation (15) simplifies to a computationally efficient scalar formula for updating the individual residuals

$$r_{(i)} = r_i/(1 - h_i). \tag{18}$$

h_i is often referred to as the *leverage value* associated with the i-th sample (row) in $\mathbf{X}$. For $\hat{y}_{i,(i)}$ denoting the prediction of the i-th sample after deleting it from the regression modelling problem in (1), the LooCV *PRESS*-statistic, is given by

$$PRESS = \sum_{i=1}^{n}(y_i - \hat{y}_{i,(i)})^2 = \sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i - 1/n}\right)^2. \tag{19}$$

In (19) $\hat{y}_i$ is the i-th entry in the vector of fitted values $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}_{OLS} + b_0$, and h_i denotes the i-th diagonal element of the projection matrix $\mathbf{H}$ defined in (4) above. The denominator $(1 - h_i - 1/n)$ scales the i-th model residual $(y_i - \hat{y}_i)$ to obtain the exact LooCV prediction residual ($y_i - \hat{y}_{i,(i)}$). The term $1/n$ in this denominator accounts for the centring of the $\mathbf{X}$-columns and the associated inclusion of a constant term (b_0) in the regression model (3).

From the last identity in Equation (4) it is clear that the entries of the n-vector $\mathbf{h} = [h_1\ h_2\ \ldots\ h_n]'$, corresponding to the diagonal elements of $\mathbf{H}$, are identical to the square of the norms of the $\mathbf{T}$-rows, i.e.

$$\mathbf{h} = (\mathbf{T} \odot \mathbf{T})\mathbf{1}. \tag{20}$$

Here, $\mathbf{T} \odot \mathbf{T}$ denotes the Hadamard (element-wise) product of $\mathbf{T}$ with itself and $\mathbf{1} \in \mathbb{R}^r$ is the constant vector with 1's in all entries. Appropriate choices of the matrix $\mathbf{T}$ can be obtained in various ways including both the QR-factorisation and the SVD of $\mathbf{X}$.

It should be noted that calculating the matrix inverse $(\mathbf{X}'\mathbf{X})^{-1}$ in the process for finding the diagonal $\mathbf{h}$ of $\mathbf{H}$ in (4) is neither required nor recommended in practice. In general, the explicit calculation of matrix inverses (for non-diagonal matrices) should be avoided whenever possible due to various unfavourable computational aspects, see Björck [17, Section 1.2.6].

2) THE GENERALISED CROSS-VALIDATION

The $GCV(\lambda)$ was proposed by Golub et al. [4] as a fast method for choosing good regularisation parameter (λ) values in RR. Here, we consider the definition

$$\begin{aligned} GCV(\lambda) &\overset{\text{def}}{=} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_{\lambda,i}}{1 - \bar{h}_\lambda - 1/n} \right)^2 \\ &= (1 - df(\lambda)/n)^{-2} \|\mathbf{y} - \mathbf{X}\mathbf{b}_\lambda\|^2, \end{aligned} \tag{21}$$

where $(y_i - \hat{y}_{\lambda,i})$ is the i-th entry of the residual vector $\mathbf{r}_\lambda = \mathbf{y} - \hat{\mathbf{y}}_\lambda$, $\bar{h}_\lambda \overset{\text{def}}{=} \frac{1}{n}\sum_{j=1}^{r} \frac{s_j}{s_j + \lambda/s_j}$ and the effective degrees of freedom $df(\lambda) \overset{\text{def}}{=} n\bar{h}_\lambda + 1$. This definition of $GCV(\lambda)$ is proportional (by the sample size n) to the definition given in [4, page 216]. The $GCV(\lambda)$ is explained as a rotation invariant alternative to the LooCV that provides an approximation of the $PRESS(\lambda)$-statistic defined below.

From the elementary matrix-vector multiplication formula (36) for computing the regression coefficients $\mathbf{b}_\lambda$, it is clear that $GCV(\lambda)$ can be calculated very efficiently for a large number of different λ-values once the non-zero singular values of $\mathbf{X}$ are available.

In their justification of $GCV(\lambda)$ as the preferable choice over the exact LooCV-based $PRESS(\lambda)$, Golub and co-workers stressed the unsatisfactory properties of the $PRESS$-function when the rows of $\mathbf{X}$ are exactly or approximately orthogonal. In this case, the estimated regression coefficient $\mathbf{b}_\lambda^{(i)}$ (obtained by excluding the i-th row $\mathbf{x}_i$ of $\mathbf{X}$) must be correspondingly orthogonal (or nearly orthogonal) to the excluded sample $\mathbf{x}_i$. Consequently, the associated leave-one-out prediction $\hat{y}_{i,(i)}(= \mathbf{x}_i\mathbf{b}_\lambda^{(i)})$ becomes a poor estimate of the corresponding i-th response value y_i.

NOTE: In situations such as the one just described, it makes little sense to think of the $\mathbf{X}$-data as a collection of independent random samples, and the statistical motivation for considering the LooCV idea becomes correspondingly inferior. In [4] it is claimed that any parameter selection procedure should be invariant under orthogonal transformations of the ($\mathbf{X}$, $\mathbf{y}$)-data. We are sceptical of this requirement as an inexpedient restriction. This relates to the context of approximating the $PRESS$-statistic for situations where a segmented/folded cross-validation approach is appropriate.

III. CALCULATION OF THE CROSS-VALIDATION BASED *PRESS*(λ)-FUNCTIONS

From Equations (17, 19) and the matrix- and vector augmentations in Equation (6), it is clear that the computationally fast versions of the SegCV and LooCV with the associated $PRESS$-statistic are also valid for TR-problems when the regularisation parameter λ is treated as a fixed quantity.

Below we will first handle the general case of segmented cross-validation. Thereafter we derive an equation assuring fast calculations of the regularised leverages in the vectors $\mathbf{h}_\lambda$ necessary for the LooCV situation. The required calculations are remarkably similar to a computationally efficient calculation of the fitted values $\hat{\mathbf{y}}_\lambda$ and closely related to the corresponding regularised regression coefficients $\mathbf{b}_\lambda$ in (36). Both $\mathbf{h}_\lambda$, $\hat{\mathbf{y}}_\lambda$ (and $\mathbf{b}_\lambda$) can be obtained from the SVD of the original centred data matrix $\mathbf{X}$. This makes the computations of the exact LooCV-based $PRESS(\lambda)$-function defined in (26) below about as efficient as the approximation obtained by the $GCV(\lambda)$ in (21).

A. *EXACT PRESS*(λ)*-FUNCTIONS FROM THE SVD OF THE AUGMENTED MATRIX* $\mathbf{X}_\lambda$

Again, we assume that the centred $\mathbf{X}$ has full rank r and that $\mathbf{X} = \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r'$ is the associated compact SVD. By defining $\mathbf{S}_{\lambda,r}$ to be the diagonal $r \times r$ matrix with non-zero diagonal entries $\sqrt{s_j^2 + \lambda}$, $j = 1, \ldots, r$, the r most dominant singular values of the augmented matrix $\mathbf{X}_\lambda$ in (6) are given by the diagonal elements of $\mathbf{S}_{\lambda,r}$. From equation (34) in Section II, the right singular vectors $\mathbf{V}_r$ of $\mathbf{X}$ are also the right singular vectors of $\mathbf{X}_\lambda$, and the associated r left singular vectors are given by

$$\begin{aligned} \mathbf{T}_{\lambda,r} = \mathbf{X}_\lambda\mathbf{V}_r\mathbf{S}_{\lambda,r}^{-1} &= \begin{bmatrix} \mathbf{X}\mathbf{V}_r\mathbf{S}_{\lambda,r}^{-1} \\ \sqrt{\lambda}\mathbf{I}\mathbf{V}_r\mathbf{S}_{\lambda,r}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{U}_r\mathbf{S}_r\mathbf{S}_{\lambda,r}^{-1} \\ \sqrt{\lambda}\mathbf{V}_r\mathbf{S}_{\lambda,r}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{\lambda,r} \\ \sqrt{\lambda}\mathbf{V}_r\mathbf{S}_{\lambda,r}^{-1} \end{bmatrix}, \end{aligned} \tag{22}$$

where the matrix $\mathbf{U}_{\lambda,r} \overset{\text{def}}{=} \mathbf{U}_r\mathbf{S}_r\mathbf{S}_{\lambda,r}^{-1}$ denoting the upper n rows of $\mathbf{T}_{\lambda,r}$ is the part of actual interest (the additional left singular vectors not included in (22) are all zeros in the upper n entries). Because $\mathbf{S}_r\mathbf{S}_{\lambda,r}^{-1}$ is $(r \times r)$ diagonal, $\mathbf{U}_{\lambda,r}$ is obtained by scaling the j-th column ($1 \le j \le r$) of $\mathbf{U}_r$ with $\sqrt{s_j/(s_j + \lambda/s_j)}$.

From the above definition of $\mathbf{U}_{\lambda,r}$, calculation of the *PRESS*-residuals associated with the n original $(\mathbf{X}, \mathbf{y})$ data points in the augmented least squares problem $\mathbf{X}_\lambda \mathbf{b} = \mathbf{y}_0$ is straight forward. According to Equations (4, 22), the regularised hat matrix $\mathbf{H}_\lambda$ is given by

$$\mathbf{H}_\lambda = \mathbf{U}_{\lambda,r}\mathbf{U}'_{\lambda,r}. \tag{23}$$

For each choice of the regularisation parameter $\lambda > 0$ and the corresponding expression for the regression coefficients $\mathbf{b}_\lambda$ in Equation (36), the fitted values are

$$\begin{aligned}\hat{\mathbf{y}}_\lambda &= \mathbf{X}\mathbf{b}_\lambda + b_{0,\lambda} = (\mathbf{U}_r\mathbf{S}_r)\mathbf{c}_\lambda + b_{0,\lambda}\\ &= \mathbf{H}_\lambda \mathbf{y} + b_{0,\lambda}.\end{aligned} \tag{24}$$

Hence,

$$PRESS(\lambda) \overset{\text{def}}{=\!=} \sum_{k=1}^{K} \|[\mathbf{I}_{n_k} - \mathbf{H}_{\lambda,\{k\}} - 1/n]^{-1}(\mathbf{y}_{\{k\}} - \hat{\mathbf{y}}_{\lambda,\{k\}})\|^2, \tag{25}$$

where $\mathbf{y}_{\{k\}} - \hat{\mathbf{y}}_{\lambda,\{k\}}$ is the sub-vector of the residual vector $\mathbf{r}_\lambda = \mathbf{y} - \hat{\mathbf{y}}_\lambda$ corresponding to the k-th CV segment and $\mathbf{H}_{\lambda,\{k\}}$ is the associated sub-matrix of $\mathbf{H}_\lambda$. While Equation (25) defines the general, segmented cross-validation case, the special case of LooCV simplifies considerably. Only the diagonal entries of $\mathbf{H}_\lambda$ (the sample leverages) are required, i.e., Equation (25) simplifies to

$$PRESS(\lambda) \overset{\text{def}}{=\!=} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_{\lambda,i}}{1 - h_{\lambda,i} - 1/n}\right)^2. \tag{26}$$

Note that $\bar{h}_\lambda$ in the denominator of Equation (21) defining $GCV(\lambda)$ is identical to the mean of the $\mathbf{h}_\lambda$-entries, i.e. $\bar{h}_\lambda = (1/n)\sum_{i=1}^{n} h_{\lambda,i}$, due to the fact that $\mathbf{U}_r$ is an orthogonal matrix. Also note that the diagonal entries of $\mathbf{H}_\lambda$ can be calculated directly by

$$\mathbf{h}_\lambda = (\mathbf{U}_{\lambda,r} \odot \mathbf{U}_{\lambda,r})\mathbf{1} = (\mathbf{U}_r \odot \mathbf{U}_r)\mathbf{d}_\lambda, \tag{27}$$

where the coefficient vector $\mathbf{d}_\lambda = [d_{1,\lambda} \ \dots \ d_{r,\lambda}]' = (\mathbf{S}_r\mathbf{S}_{\lambda,r}^{-1})^2\mathbf{1} \in \mathbb{R}^r$ has the entries

$$d_{i,\lambda} = \frac{s_j^2}{s_j^2 + \lambda} = \frac{s_j}{s_j + \lambda/s_j}, \text{ for } 1 \leq j \leq r. \tag{28}$$

Consequently, the evaluation of the $PRESS(\lambda)$-function defined in (26) is essentially available at the additional computational cost of two matrix-vector multiplications (Equations (24,27)) where the matrices $(\mathbf{U}_r\mathbf{S}_r$ and $\mathbf{U}_r \odot \mathbf{U}_r)$ are fixed and the associated coefficient vectors $\mathbf{c}_\lambda$ and $\mathbf{d}_\lambda$ are obtained by elementary arithmetic operations for each choice of $\lambda > 0$. A note on the number of floating point operations (flops) required for the fast calculation of the LooCV-based $PRESS(\lambda)$-function is included in Appendix H.

B. ALTERNATIVE STRATEGIES FOR ESTIMATING THE SEGCV-BASED PRESS(λ)-FUNCTION

The LooCV calculations in the previous section can be implemented at low computational costs dominated by the SVD of $\mathbf{X}$. The SegCV version, however, also involves the inversion of several matrices associated with each combination of the regularisation parameter value of λ and cross-validation segment. In situations with many CV segments, e.g., defined by relatively small groups of replicates, the additional computational costs may be acceptable as the matrices to be inverted are small. However, for large datasets with few segments, e.g., 5-10, the required amount of computations may be rather large (comparable to explicitly holding out samples and recalculating a full TR model from scratch for each CV segment).

We therefore describe two alternative strategies for speeding up calculations. The first one is based on approximating the *PRESS*-values, while the second strategy involves clever usage of a small subset of exact $PRESS(\lambda)$-values to estimate the minimum of the $PRESS(\lambda)$-value and/or the complete $PRESS(\lambda)$ curve within some range of the regularisation parameter value.

1) $PRESS(\lambda)$ APPROXIMATED BY SEGMENTED VIRTUAL CROSS-VALIDATION – VIRCV

We will consider a faster alternative for approximating the SegCV approach for the type of situations just described. In the following, we assume (without loss of generality) that the uncentred data matrix

$$\mathbf{X} = \begin{bmatrix}\mathbf{X}_1\\ \mathbf{X}_2\\ \vdots\\ \mathbf{X}_K\end{bmatrix} \text{ together with the uncentred response vector}$$

$$\mathbf{y} = \begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \vdots\\ \mathbf{y}_K\end{bmatrix} \quad (K \geq 2) \tag{29}$$

is composed by K distinct sample segments. For $1 \leq k \leq K$, we assume that $\mathbf{U}_k\mathbf{S}_k\mathbf{V}'_k = \mathbf{X}_k$ denotes the compact SVD of segment number k, and that n_k is the number of rows in $\mathbf{X}_k$ so that the total number of samples is $n = \sum_{k=1}^{K} n_k$.

From the SVD of the k-th segment, we obtain the identity $\mathbf{U}'_k\mathbf{X}_k = \mathbf{S}_k\mathbf{V}_k$. Consequently, the orthogonal transformation performed by left multiplication with the $(n_k \times n_k)$ matrix $\mathbf{U}'_k$ transforms the samples segment $\mathbf{X}_k$ into a matrix of strictly orthogonal rows. Now we define the two block diagonal matrices

$$\mathbf{T} = \begin{bmatrix}\mathbf{U}_1 & & & \\ & \mathbf{U}_2 & & \\ & & \ddots & \\ & & & \mathbf{U}_K\end{bmatrix} \text{ and } \tilde{\mathbf{T}} = \begin{bmatrix}\mathbf{T} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}, \tag{30}$$

with the properties $\mathbf{T}'\mathbf{T} = \mathbf{T}\mathbf{T}' = \mathbf{I}$ and $\tilde{\mathbf{T}}'\tilde{\mathbf{T}} = \tilde{\mathbf{T}}\tilde{\mathbf{T}}' = \mathbf{I}$, i.e., both $\mathbf{T}$ and $\tilde{\mathbf{T}}$ are orthogonal.

The formulation of TR-modelling for uncentred $\mathbf{X}$ and explicit inclusion of the constant term corresponds to finding the least squares solution of the linear system

$$\begin{bmatrix} \mathbf{1} & \mathbf{X} \\ 0 & \sqrt{\lambda}\mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \tag{31}$$

and left multiplication of (31) by the orthogonal matrix $\tilde{\mathbf{T}}'$ yields the system

$$\begin{bmatrix} \mathbf{T}'\mathbf{1} & \mathbf{T}'\mathbf{X} \\ 0 & \sqrt{\lambda}\mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \mathbf{T}'\mathbf{y} \\ \mathbf{0} \end{bmatrix}. \tag{32}$$

Note that the associated normal equations of the systems in (31) and (32) are identical. Hence, their least squares solutions are also identical.

Definition of the segmented virtual cross-validation We define the *segmented virtual cross-validation (VirCV)* strategy as the process of applying the LooCV strategy to the transformed system in equation (32). As is noted above, multiplication by $\mathbf{T}'$ has the effect of orthogonalising the rows within each of the K segments in the $\mathbf{X}$ matrix.

The heuristic argument for justifying the VirCV approach as an approximation of a SegCV approach is that the rows within each transformed data segment are unsupportive of each other under the LooCV strategy (due to the internal "decoupling" of each segment into a set of mutually orthogonal row vectors). However, from practical cases, it can be observed that the accuracy of this approximation depends on the level of similarity between the original samples within each segment of data points.

Note that contrary to the LooCV, the *GCV* is not useful in combination with the VirCV strategy. The reason for this is that the singular values of $\mathbf{X}$ are invariant under orthogonal transformations. From equation (21) and the definition of $\bar{h}_\lambda$ it follows that $GCV(\lambda)$ is also invariant under orthogonal transformations, i.e., the systems in (31) and (32) lead to the same $GCV(\lambda)$-function.

With the VirCV we are clearly cross-validating on the orthogonal phenomena caused by the samples within each segment. As all the samples in a segment contribute to identifying these directions, the VirCV cannot be expected to provide exactly the same results as the SegCV. One may, however, expect that when the different segments are carefully arranged to contain highly similar samples only (which is a reasonable assumption to make for most organised studies with such data segments), then the VirCV should provide a useful approximation to the SegCV. This will be demonstrated in the application section below. For special situations deviating from highly similar samples in the segments, see Appendix D.

Computational aspects in the leverage corrections for the VirCV As is noted in association with (29), the VirCV procedure requires an initial calculation of the transformation $\mathbf{T}$ from the segments of the uncentred $\mathbf{X}$-data. For a correct implementation of the computational shortcuts similar to those of the LooCV, it is necessary to mean centre the data matrix $\mathbf{X}$ prior to executing the $\mathbf{T}$-transformation and the least squares modelling. In practice, one must therefore mean centre the data prior to the multiplication with $\mathbf{T}'$ (or, equivalently, one can multiply by $\mathbf{T}'$ and subtract the projection of the transformed data onto the transformed vector $\mathbf{T}'\mathbf{1}$ of ones). As $\mathbf{T}$ is an orthogonal transformation the angles and in particular the orthogonality between vectors will be preserved. For the transformed data, modelling by including a constant term is therefore associated with the transformed vector $\mathbf{T}'\mathbf{1}$ of ones. With $\mathbf{X}_c$ and $\mathbf{y}_c$ denoting the centred data matrix and the associated centred response vector, respectively, the vector $\mathbf{T}'\mathbf{1}$ is orthogonal to the columns of the transformed centred data $\mathbf{T}'\mathbf{X}_c$ and $\|\mathbf{T}'\mathbf{1}\| = \|\mathbf{1}\| = \sqrt{n}$. The justification for the leverage correction described earlier therefore still holds, but the particular correction terms $(1/n)$ changes.

With the transformed centred predictors $\tilde{\mathbf{X}} = \mathbf{T}'\mathbf{X}_c$ and responses $\tilde{\mathbf{y}} = \mathbf{T}'\mathbf{y}_c$ in (32), the associated fitted values as $\hat{\tilde{\mathbf{y}}}_\lambda = \tilde{\mathbf{X}}\mathbf{b}_\lambda$, the *PRESS*-function for the VirCV is given by

$$\begin{aligned} PRESS_{VirCV}(\lambda) &= \sum_{i=1}^{n} (\tilde{y}_i - \hat{\tilde{y}}_{\lambda,i,-1})^2 \\ &= \sum_{i=1}^{n} \left(\frac{\tilde{y}_i - \hat{\tilde{y}}_{\lambda,i}}{1 - h_{\lambda,i} - m_i/n} \right)^2 . \end{aligned} \tag{33}$$

Here the leverages $h_{\lambda,i}$ are calculated as in (27) based on the transformed version $\tilde{\mathbf{X}}$ of the centred data, and the enumerator of the correction terms are the entries of the vector $\mathbf{m} = \mathbf{T}'\mathbf{1} \odot \mathbf{T}'\mathbf{1} \in \mathbb{R}^n$. This means that the correction term $1/n$ in the denominator of (26) must be replaced by m_i/n in (33), where m_i denotes the i-th entry of the vector $\mathbf{m}$ (to be consistent with the orthogonal transformation of the regularised least squares problem).

A comparison of the number of flops required for the VirCV compared to the SegCV is included in Appendix I.

2) APPROXIMATED *PRESS*-FUNCTION USING SUBSETS OF λ

Minimum *PRESS*-value estimation

If TR is used in an automated system (without subjective assessment) or only the optimal $PRESS(\lambda)$ is needed, we can avoid redundant calculations by searching for the λ value that minimises (25) instead of calculating a large range of solutions. A possible approach for such a search can be based on the golden section search with parabolic interpolation [18]. This method performs a search for the minimal function value over a bounded interval of a single parameter. To leverage the previously described efficient computations of fitted values, $\hat{\mathbf{y}}_\lambda$, coefficient vectors, $\mathbf{d}_\lambda$, etc. the search for minimum $PRESS(\lambda)$ is then performed over a fixed set of λ-values. The grid of λ-values can have high resolution while still achieving a considerable advantage in computational speed compared to the exhaustive *PRESS*-function calculations. It is well-known that this type of function minimisation cannot guarantee the optimal value to be found, however, the *PRESS*-functions of interest often have relatively smooth and simple graphs, where a global minimum over the λ-interval of interest can be found with high accuracy.

***PRESS*(λ)-function estimation by spline interpolation**

In cases where estimating the detailed $PRESS(\lambda)$-function is beneficial, e.g., for plotting and inspection, it may be possible to reduce the number of accurate $PRESS(\lambda)$-evaluations to be calculated quite substantively without sacrificing much precision in the estimation.

We propose a cubic spline strategy, where the $PRESS(\lambda)$-function is estimated from a small set of distinct λ-values, and new values are added to the set iteratively until the difference between estimation and true *PRESS*-value falls below a chosen threshold for all λ-values in the extended set. The latter is determined by cross-validation of the cubic spline interpolation, i.e., a low-cost operation.

As with the *PRESS*-minimisation procedure, we consider a fixed set of λ-values from which we choose starting points and select subsequent values. The λ-values extending the set in each iteration are the ones halfway to neighbours of the chosen λ-values on both sides, effectively doubling the local density of λ-values where needed (low accuracy of spline approximation). Starting values for the initial set of λs can be chosen equidistant (on a $\log_{10}$ scale) or the sequence obtained using the above "Minimum *PRESS*-value estimation" strategy. Experience with real datasets indicates that the latter is an efficient strategy that may provide close to exact estimation of the minimum *PRESS*-value.

C. A SHORT NOTE ON MODEL SELECTION HEURISTICS

With the key formulas derived above we obtain efficient model selection procedures from minimising the $PRESS(\lambda)$- or the $GCV(\lambda)$-functions with respect to the regularisation parameter λ. However, the minima of these functions will not necessarily assure the selection of the best model in terms of future predictions. This is particularly the case when the *PRESS*- and *GCV* functions are relatively flat for a relatively large interval of λs containing the minimum value. In such situations it is often useful to invoke heuristic principles such as *Occam's razor* for identifying a simpler model (in terms of the norm of the regression coefficients) at a small additional cost in terms of the *PRESS* (or the *GCV*):

The '**1 standard error rule**' described in [1] obtains a simpler (more regularised) alternative by selecting a model where the *PRESS*-statistic is within one standard error of the *PRESS*-minimal model. More precisely, we first identify the minimum *PRESS* value and calculate the standard error of the squared cross-validation errors associated with this model. Then the largest regularisation parameter value where the associated model has a *PRESS*-statistic within one standard error of the *PRESS*-minimum is selected.

The 'χ^2 **model selection rule**' to determine the regularisation parameter was originally introduced for model selection with Partial Least Squares regression modelling [19]. By assuming that the residuals associated with the minimum value $PRESS_{min}$ of $PRESS(\lambda)$ are randomly drawn from a normal distribution, the statistic given by $n \cdot PRESS_{min}/\sigma^2$, where σ^2 is the associated (unknown) variance, follows a χ^2_n distribution (where n is the degrees of freedom). By fixing a particular significance level α, the selection rule says: *Choose the largest possible value of* λ *so that* $n \cdot PRESS_{min}/PRESS(\lambda) \geq \chi^2_{n,\alpha}$. Here, $\chi^2_{n,\alpha}$ is the lower α-quantile of the χ^2_n distribution and $PRESS(\lambda)$ is a substitute for σ^2.

Based on the efficient formulas for calculating the $PRESS(\lambda)$ function, both these model selection alternatives can be implemented without affecting the total computational costs significantly.

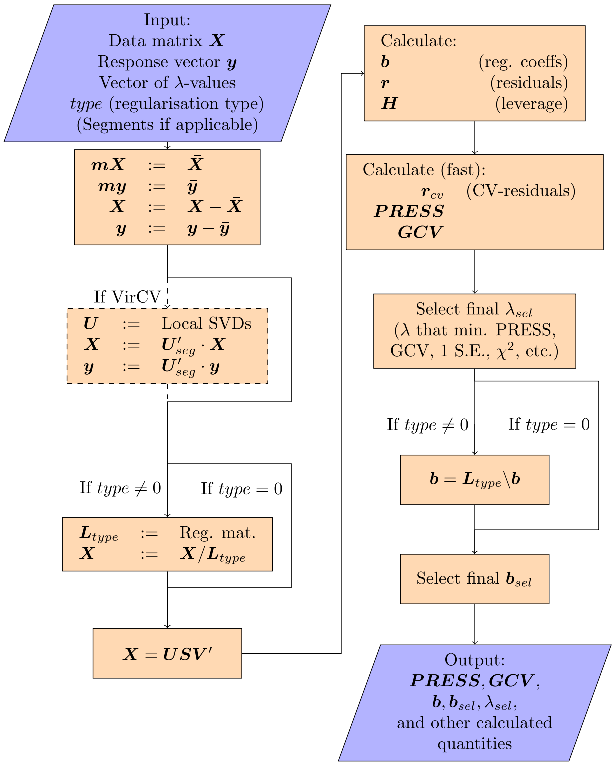
Input:
Data matrix $\boldsymbol{X}$
Response vector $\boldsymbol{y}$
Vector of λ-values
type (regularisation type)
(Segments if applicable)
$\boldsymbol{mX} := \bar{\boldsymbol{X}}$
$\boldsymbol{my} := \bar{\boldsymbol{y}}$
$\boldsymbol{X} := \boldsymbol{X} - \bar{\boldsymbol{X}}$
$\boldsymbol{y} := \boldsymbol{y} - \bar{\boldsymbol{y}}$
If VirCV
$\boldsymbol{U}$:= Local SVDs
$\boldsymbol{X} := \boldsymbol{U}'_{seg} \cdot \boldsymbol{X}$
$\boldsymbol{y} := \boldsymbol{U}'_{seg} \cdot \boldsymbol{y}$
If $type \neq 0$
If $type = 0$
$\boldsymbol{L}_{type}$:= Reg. mat.
$\boldsymbol{X} := \boldsymbol{X}/\boldsymbol{L}_{type}$
$\boldsymbol{X} = \boldsymbol{USV}'$
Calculate:
$\boldsymbol{b}$ (reg. coeffs)
$\boldsymbol{r}$ (residuals)
$\boldsymbol{H}$ (leverage)
Calculate (fast):
$\boldsymbol{r}_{cv}$ (CV-residuals)
$\boldsymbol{PRESS}$
$\boldsymbol{GCV}$
Select final λ_{sel}
(λ that min. PRESS, GCV, 1 S.E., χ^2, etc.)
If $type \neq 0$
If $type = 0$
$\boldsymbol{b} = \boldsymbol{L}_{type} \backslash \boldsymbol{b}$
Select final $\boldsymbol{b}_{sel}$
Output:
$\boldsymbol{PRESS}, \boldsymbol{GCV}$,
$\boldsymbol{b}, \boldsymbol{b}_{sel}, \lambda_{sel}$,
and other calculated quantities

FIGURE 1. Flow chart illustrating the LooCV, segmented CV, and VirCV. Most of the steps are common to all the algorithms. For the virtual CV we calculate local SVDs for each segment and left multiply by the transposed left-singular vectors of the segments prior to applying the regularisation matrix (if any). Detailed calculations and minor differences between the algorithms such as the modified leverage correction for VirCV are not shown.

IV. APPLICATIONS

In the following, we demonstrate some applications of our fast cross-validation approaches for model selection within the TR framework for several real-world datasets. We consider situations where both leave-one-out and segmented cross-validation are appropriate. The required algorithms were implemented and executed in MATLAB, and prototype code is given in Appendices E-G. A corresponding implementation in R-code will be made available upon publication at https://CRAN.R-project.org/package=TR. We used a computer running Mac OS Ventura 13.0.1 and MATLAB R2022a, with 16 GB RAM, and an M1 Pro 10-core processor. For the

TABLE 1. ***Octane data. MSE*** **(from test data) using various regularisation types and parameter selection methods.**

Regularisation type Parameter selection method	L_2	First derivative	Second derivative
Minimum PRESS value	0.057	0.047	0.038
Minimum GCV value	0.057	0.047	0.039
PRESS and 1 standard error rule	0.059	0.045	0.036
PRESS and χ^2-rule	0.073	0.047	0.039

TABLE 2. ***Pork fat data. MSE*** **(from test data) for the SFA response using various regularisation types and parameter selection methods.**

Regularisation type Parameter selection method	L_2	First derivative	Second derivative
Minimum PRESS value	4.46	5.39	5.56
Minimum GCV value	4.36	5.45	5.58
PRESS and 1 standard error rule	4.58	5.56	5.72
PRESS and χ^2-rule	4.11	4.32	4.20

derivative regularisation, we use the full rank approximations described in Section C with the scaling coefficient set to $\epsilon = 10^{-10}$ in the appended rows in the discrete regularisation matrices. This is done to mitigate the numerical impact from these rows in the resulting regression coefficients.

A. THE FAST LEAVE-ONE-OUT CROSS-VALIDATION

1) DATASETS

The following datasets will be considered in the examples presented below:

1) *Octane data* [20]. This dataset consists of near-infrared (NIR) spectra of gasoline. There are 60 samples and 401 features (wavelengths in the range $900\,nm - 1700\,nm$). The response value is the octane number measured for each sample.
2) *Pork fat data* [21]. This dataset consists of Raman spectra measured on pork fat tissue. There are 105 samples, 5567 features (wavenumbers in the range $1889.9\,cm^{-1} - 200.1\,cm^{-1}$), and 19 different responses. For modelling and prediction, we only consider the response consisting of saturated fatty acids as a percentage of total fatty acids, hereafter referred to as SFA.
3) *Prostate gene data* [22]. The dataset is a microarray gene expression dataset. There are 102 samples, and the gene expression of 12600 different genes were measured. The response is binary (cancer/not cancer), and we consider the dummy-regression approach to the underlying classification problem. For this dataset, we standardise the data prior to modelling. The standardisation will introduce a small bias in the model selection that will be discussed later.

For all datasets, we have used approximately $2/3$ of the available samples for model building and -selection. The remaining $1/3$ of the samples were used for testing the selected models. (Note that our choice of data splitting is somewhat arbitrary, just to serve the purpose of illustrating the ideas with an appropriate number of samples for both training and testing.) We considered the following model selection alternatives identifying good regularisation parameter candidates: (i) $PRESS_{min}$ – the minimum $PRESS(\lambda)$-value, (ii) GCV_{min} – the minimum $GCV(\lambda)$-value, (iii) the 1 standard error rule for $PRESS(\lambda)$, (iv) the χ^2-rule for $PRESS(\lambda)$ using the significance level $\alpha = 0.2$.

2) MODEL SELECTION AND PREDICTION

For each dataset, the modelling was based on a grid search of 1000 regularisation parameter candidate values spaced uniformly on a log-scale. For the octane data, the displayed values were in the range 10^{-4} to 10^5, for the Pork fat data in the range 10^2 to 10^{25}, and for the Prostate data in the range 10^{-1} to 10^8. Different ranges were chosen for each dataset to avoid irrelevant levels of regularisation, and to obtain a good visualisation of the *PRESS*- and *GCV* curves including the located minima. In Figures 2–4 the $PRESS/n$ and GCV/n are plotted as functions of the regularisation parameter for the different datasets and the different choices of the regularisation matrix. Such plots are useful for model selection as they allow for a direct comparison of the model quality for different values of the regularisation parameter. Division of the *PRESS*- and *GCV* values by the sample size n makes the model selection statistics directly comparable to the prediction results obtained by the test sets. The test set results are shown in the Tables 1–3.

For the prostate data, the percentage correctly classified on the training set using cross-validation (classifying each sample to the largest of the fitted target values when using $0/1$ dummy-coding for the group memberships) is 91.2% for all the parameter selection methods (it should be noted that this number happens to be identical to the test set result for most of the parameter selection methods).

It should be noted that most of the displayed *PRESS*- (and *GCV*-) curves are relatively flat without a very distinct minimum point. Therefore it may be advantageous to employ either the 1 S.E. rule or the χ^2-rule to assure the selection of a simpler model. For the Prostate data, in particular, we note that the smallest available candidate regularisation parameter value provides the minimum *PRESS*-value. The effect in terms of prediction when using the 1 S.E. rule or the χ^2-rule to obtain a simpler model varies between the datasets. For the Pork fat data, the χ^2-rule gives better prediction than the other parameter selection methods for the SFA response, while the χ^2-rule selects a poorer model than the other parameter selection methods on the Prostate data.

For the most precise identification of the *PRESS*- and *GCV*-minima a numerical optimiser should be used. However, in most practical situations the suggested strategy of considering just a subset of candidate regularisation parameter values is usually good enough for approximating the minima before doing the subsequent identification of parsimonious models (based on the principle of Occam's razor) that predict well.

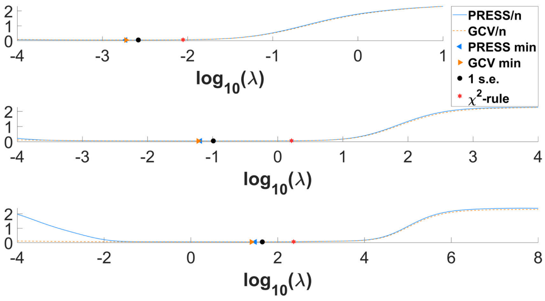

FIGURE 2. ***Octane data. PRESS/n* and *GCV/n* for a range of regularisation parameter values and different regularisation matrices. *Top*: L_2 regularisation. *Middle*: 1st derivative regularisation. *Bottom*: 2nd derivative regularisation. The minimum *PRESS* and *GCV* values have been marked, as well as the regularisation parameter values selected by the 1 S.E. rule and the χ^2-rule.**

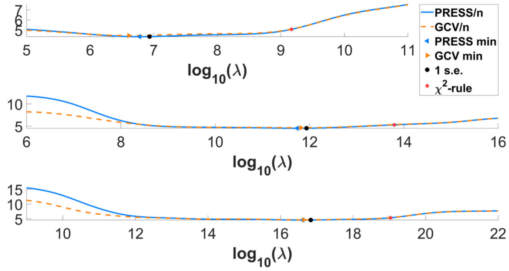

FIGURE 3. ***Pork fat data and SFA response. PRESS/n* and *GCV/n* for a range of regularisation parameter values and different regularisation matrices. *Top*: L_2 regularisation. *Middle*: 1st derivative regularisation. *Bottom*: 2nd derivative regularisation. The minimum *PRESS* and *GCV* values have been marked, as well as the regularisation parameter values selected by the 1 S.E. rule and the χ^2-rule.**

3) REGRESSION COEFFICIENTS

Figure 5 shows the octane data together with the *PRESS*-minimal regression coefficients using the L_2-, the first derivative-, and the second derivative regularisations. Note that the choice of regularisation matrix heavily influences the appearance of the regression coefficients without the minimum *PRESS*- or *GCV* values changing much. Table 1 confirms that the predictive powers are relatively similar for all these models. Doing consistent model interpretations solely based on the regression coefficients in Figure 5 is obviously a challenging (if not impossible) task, see also [23].

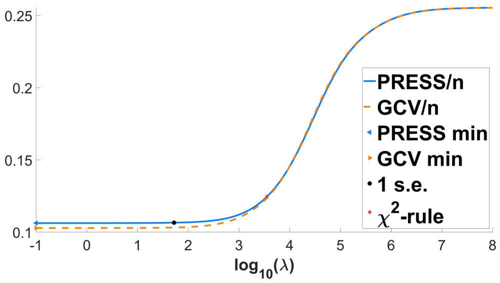

FIGURE 4. ***Prostate data. PRESS/n* and *GCV/n* for a range of regularisation parameter values using L_2 regularisation. The minimum *PRESS* and *GCV* values have been marked, as well as the regularisation parameter values selected by the 1 S.E. rule and the χ^2-rule.**

TABLE 3. ***Prostate data.* Percentage of correctly classified (PCC) samples using the test set predictions of the selected 0 − 1 dummy regression model based on L_2 regularisation.**

Parameter selection method	PCC test set
Minimum PRESS value	91.2
Minimum GCV value	91.2
PRESS and 1 standard error rule	91.2
PRESS and χ^2-rule	88.2

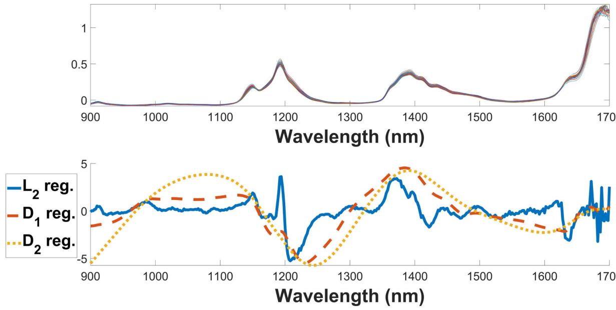

FIGURE 5. ***Octane data. Top*: Plot of the NIR spectra of octane. *Bottom*: *PRESS*-minimal regression coefficients based on different regularisation matrices.**

4) COMPUTATIONAL SPEED

Table 4 shows the computation times for model selection with the different datasets and different types of regularisation when varying the number of regularisation parameter candidate values (varying the number of points in the search grid). The times in Table 4 also include the computation of the regression coefficients corresponding to the minimal *GCV* and *PRESS* values for all responses. The main differences in computational time between finding the SVD in the case of L_2 regularisation and in the cases of first- and second-derivative regularisation are due to the initial calculations of $\tilde{\mathbf{X}}$, see Section C. Similarly, the required transformation of the regression coefficients (see (50)) explains the increase in computational time from calculating the SVD only to finding *PRESS*, *GCV* and regression coefficients for a single regularisation parameter value for the first and second derivative regularisation.

B. SEGMENTED CROSS-VALIDATION

1) DATASETS

In the following we will demonstrate the use of segmented cross-validation with L_2 regularisation for three datasets:

1) Raman spectra of fish oil [24]. The dataset consists of 42 sample segments including 3 replicate spectra of each unique sample giving a total of 126 rows and 2801 wavenumbers in the range 3200 cm^{-1} to 400 cm^{-1}. The response variable was the iodine value (the response values were identical across each segment), which is frequently used as an indicator of the degree of unsaturation of fat [24]. The spectra of this dataset are plotted in Figure 6 after applying Extended Multiplicative Signal Correction (EMSC) [25] with 6th-order polynomial baseline correction.
2) Fourier transform infrared (FTIR) spectra of hydrolysates from various mixtures of rest raw materials and enzymes [26]. The dataset consists of 332 samples including 1 to 12 replicates of each unique sample giving a total of 885 rows and 571 wavenumbers in the range 1800 cm^{-1} to 700 cm^{-1}. The response variable was average molecular weight (AMW) (identical across each replicate set), which can be used as a proxy for the degree of hydrolysation. The spectra of this dataset are plotted in Figure 7.
3) Raman milk spectra [27], [28], [29]. The dataset consists of 232 unique sample segments including between 6 and 12 replicate measurements of each unique sample giving a total of 2682 rows and 2981 wavenumbers in the range 3100 cm^{-1} to 120 cm^{-1}. The response variables were the iodine value and the concentration of conjugated linoleic acid (CLA). Also for this dataset, the response values were identical across each segment. The spectra of this dataset are plotted in Figure 8 after applying EMSC with 6th-order polynomial baseline correction.

For all datasets, we have excluded the endpoint regions of the original spectra due to noise and the poor quality of the measurements. The wave numbers reported above are those included after this truncation. Approximately 2/3 of the replicate segments were used for model building and -selection, and the remaining 1/3 of the segments were used as a test set.

The following four model selection strategies were considered: (i) $PRESS_{min}$ – the minimum $PRESS(\lambda)$-value from LooCV (ignoring the presence of sample segments), (ii) GCV_{min} – the minimum $GCV(\lambda)$-value, (iii) the $PRESS_{min}$ from the SegCV (successively holding out the entire sample segments), and (iv) the $PRESS_{min}$ from the VirCV. We have chosen to focus only on the parameter selections associated with the minima of the various error curves in this part of our study (neither the χ^2-rule nor the 1 S.E. rule turned out to affect the model selections much). Neither of the two strategies for quicker estimation of *PRESS*-values is shown in the plots as the minimum *PRESS*-value (from searching) coincides with the minimum-*PRESS* value from the 1000 sampled λ-values and the cubic spline interpolation is visually indistinguishable from the full *PRESS* curve obtained from explicit segment removal.

2) FISH DATA – EFFECT OF PRE-PROCESSING

Spectroscopic measurements may be corrupted by both additive and multiplicative types of noise. Pre-processing of such data prior to modelling is therefore usually required. It is therefore of particular interest also to investigate how the model selection strategies considered above compare for pre-processed data. In particular, we will consider the Extended Multiplicative Signal Correction (EMSC) [25] with replicate corrections [30].

In general, the goal of the EMSC pre-processing is to adjust all the measured spectra to a common scale and to eliminate the possible effects of additive noise. This includes the estimation of an individual scaling constant for each spectrum and an orthogonalisation step that de-trends the spectra with respect to some set of lower-order polynomial trends (the reader is referred to the provided references for the technical details). In the present examples with Raman spectra, the samples were orthogonalised with respect to the subspace including all polynomial trends up to the 6-th degree.

The Raman spectra of fish samples were subjected to EMSC pre-processing to compensate for different scaling and competing phenomena such as fluorescence and optical/scattering effects in the equipment and samples. For the milk data, the spectrum having the least fluorescence background was chosen as a reference, though the effect of choice of reference spectrum is minimal.

For datasets including segments of replicated measurements, a replicate correction step is often considered to alleviate the presence of inter-replicate variance. Such correction can be done by an initial EMSC-based pre-processing of the spectra in each sample segment. Thereafter, the corrected sample segments can be individually mean-centred and organised into a full data matrix.

As we expect the dominant right singular vectors of the full matrix to account for the most dominant inter-replicate variance, orthogonalisation of the data with respect to one or more of the associated dimensions contributes to making the replicates more similar, see [30] for details. Because every sample in the training dataset is included in the pre-processing, some bias affecting the subsequent *PRESS*-calculations and model selection must be expected.

TABLE 4. ***Computing time*** **(in seconds) for model selection including finding the *PRESS*- and *GCV*-minimal regression coefficients when varying the number of candidate regularisation parameter values. The times are the averages of 50 repeated runs rounded to the two most significant digits.**

Number of λ-values Data (reg. type)	0 (SVD only)	1	10	100	1000	10000
Octane (L_2)	0.0014	0.0014	0.0014	0.0016	0.0024	0.013
Octane (1st derivative)	0.0034	0.0046	0.0051	0.0052	0.0055	0.017
Octane (2nd derivative)	0.0048	0.0074	0.0082	0.0082	0.0087	0.020
Pork fat (L_2)	0.018	0.023	0.023	0.026	0.040	0.26
Pork fat (1st derivative)	0.096	0.22	0.22	0.22	0.24	0.46
Pork fat (2nd derivative)	0.23	0.59	0.60	0.62	0.64	0.85
Prostate (L_2)	0.038	0.072	0.077	0.078	0.078	0.11

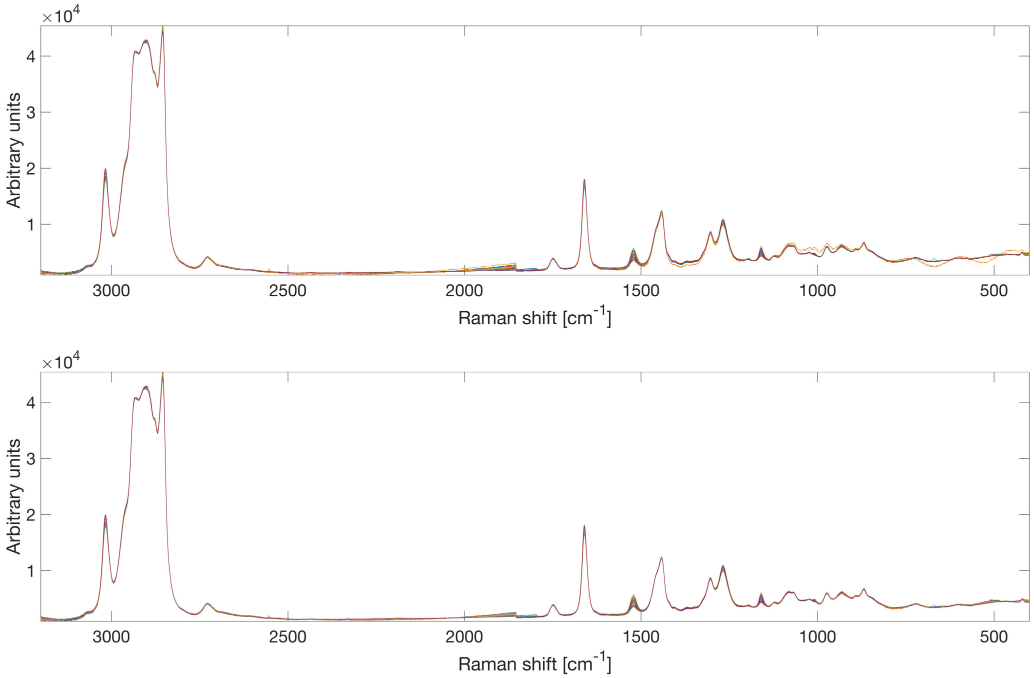

FIGURE 6. ***Plot of the fish oil spectra*** **after pre-processing with EMSC with 6th order polynomial baseline (*top*) and additional replicate correction (*bottom*).**

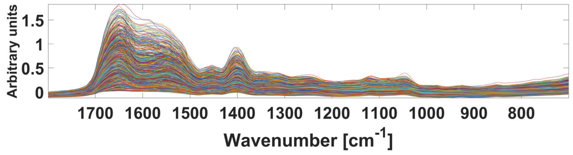

FIGURE 7. ***Plot of the hydrolysis spectra*** **after pre-processing with EMSC with 2nd order polynomial baseline.**

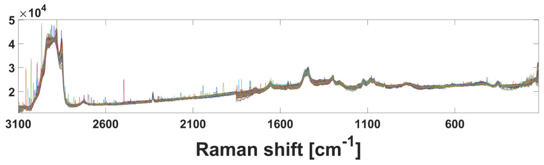

FIGURE 8. ***Plot of the milk spectra*** **after pre-processing with EMSC with 6th order polynomial baseline. Noise in some replicates is clearly visible as spikes around the main variation.**

Figure 9 shows the model selection for pre-processed fish oil data based on the pure EMSC and for the EMSC where 30% of the inter-replicate variance is removed. It is evident that the SegCV and the VirCV become considerably more similar in the latter case. As one should expect, the *GCV*- and *PRESS* curves based on the LooCV seem to provide unrealistically low error values and the selection of lesser regularised models. This phenomenon does not occur with the SegCV where an entire segment of replicates is held out in each cross-validation step. The VirCV seems quite robust against the inter-replicate variance.

The prediction results for the test set of the fish oil data with the various pre-processing alternatives are presented in Table 5, and shows that the best results are obtained with the ordinary EMSC pre-processing and model selection based on the SegCV. By simultaneously considering Figure 9, it is clear that the more heavily regularised among the selected models (those based on the largest regularisation parameter values) perform better on the test set. With standard EMSC pre-processing the minima of the VirCV is located at a smaller regularisation parameter value than for the SegCV, suggesting an explanation of the difference in predictive performance.

TABLE 5. **Fish oil data. *MSE* (from test data) for different model selection strategies and different pre-processing alternatives.**

Pre proc. \ Selection curves	LooCV	GCV	VirCV	SegCV
Raw data	20.3	21.5	12.3	9.7
EMSC	14.4	15.1	6.9	4.5
EMSC + 30% inter-replicate variance removed	14.4	15.9	6.7	6.7

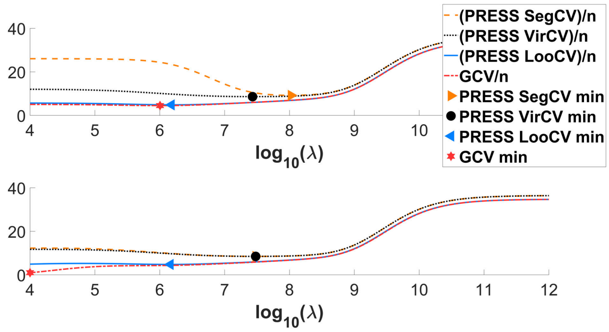

FIGURE 9. ***Fish oil data.*** **Model selection for data pre-processed with the EMSC both with and without replicate correction. Top: Standard EMSC pre-processing. Bottom: EMSC with 30% of the inter-replicate variance removed.**

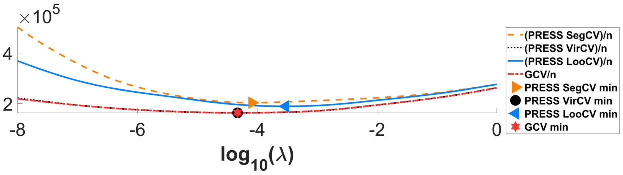

FIGURE 10. ***Hydrolysis data.*** **Different model selection strategies for a range of regularisation parameter values using 2nd derivative regularisation.**

TABLE 6. **Hydrolysis data. *MSE* (from test data) using EMSC for pre-processing.**

Pre proc. \ Selection curves	LooCV	GCV	VirCV	SegCV
EMSC	1.85	1.92	1.92	1.89

For the milk data, the prediction error estimates obtained after pre-processing the data are similar for all the parameter selection methods (table omitted), as was also the case with the raw data.

3) HYDROLYSIS DATA – HETEROGENEOUS SEGMENTS

The hydrolysis data is used as an example of a model comparison which is often performed using 5-fold or 10-fold segmented cross-validation. For the FTIR data, we have chosen a 5-fold strategy where replicates are kept together inside each fold to prevent information bleeding by replicates of the same sample appearing in both training and test data. The resulting cross-validation segments vary in size from 103 to 117 samples, each, due to the present replicate sets. We have chosen to combine this with a 2nd derivative regularisation.

In Figure 10, we have plotted the *PRESS*-curves for SegCV, VirCV, LooCV and GCV. For these highly heterogeneous cross-validation segments, the virtual cross-validation strategy coincides with *GCV*, both underestimating the prediction errors. Also, LooCV underestimates the errors, but less so. Since the general forms of the *PRESS*-curves are quite similar, the minimum *PRESS*-values are located quite close together, suggesting that for the FTIR dataset, any of the strategies will give a reasonable estimate of the optimal λ-value. As Table 6 suggests, performance when applying the regressions corresponding to minimal *PRESS*-values on the test data are also similar with a slight advantage to the more regularised LooCV solution.

4) MILK DATA – EFFICIENCY WITH MANY SEGMENTS

The milk data is an example of relatively many samples (2682) and replicate groups (232), which can be challenging with regard to computational resources when cross-validating over a large range of λ-values. As can be observed from Figure 11, the differences between SegCV, VirCV, LooCV and GCV are small both with regard to the shape of the curves and location of respective minimum values. This is due to the low variation between samples within each replicate group, in sharp contrast to the FTIR dataset with its highly heterogeneous cross-validation segments. Of more interest, is the time usage for the various strategies, which is summarised in Section IV-B5 below.

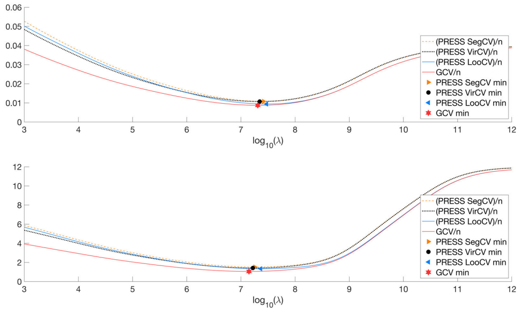

FIGURE 11. ***Milk data.*** **Different model selection strategies for a range of regularisation parameter values using L_2 regularisation. *Top*: CLA. *Bottom*: Iodine value.**

5) APPROXIMATIONS OF *PRESS*-VALUES - COMPUTATIONAL SPEED

Table 7 shows the computational times for the different model selection strategies. Both the *PRESS*- and the *GCV* values are included as computing only one of them takes approximately the same time as computing both. Because the size of the replicate segments are relatively small for the Raman datasets (3 replicate measurements for the fish oil data and 6 to 12 replicate measurements for the milk data), the SVDs

TABLE 7. Computational time for different model selection strategies for the fish oil data, hydrolysis data and milk data when considering 500 candidate regularisation parameter values. The times are given in seconds, rounded to two significant digits, and is the average of 50 repeated runs. The speedup relative to SegCV is shown in parenthesis.

Dataset	SegCV	ImpCV	VirCV	MinSearch	Spline	PRESS&GCV
Fish oil	0.624 (1)	0.100 (1/6)	0.016 (1/38)	0.015 (1/42)	0.062 (1/10)	0.010 (1/65)
Hydrolysis	0.686 (1)	0.927 (1/0.7)	0.106 (1/6)	0.118 (1/6)	0.252 (1/3)	0.096 (1/7)
Milk	867.8 (1)	8.9 (1/98)	3.3 (1/266)	3.2 (1/271)	4.1 (1/214)	2.95 (1/294)

required for the internal orthogonalisations of the segments contribute insignificantly to the total computational load. The amount of computations required for model selection based on the VirCV is therefore quite comparable to the computations required for the LooCV version of *PRESS* (and for the *GCV*). The strategy of searching for the minimum *PRESS*-value by golden section search and parabolic interpolation (MinSearch), is remarkably similar to VirCV in time usage. However, there is a trade-off between obtaining an estimate of the exact minimum value (MinSearch) and a full *PRESS*-curve (VirCV). Approximation of the SegCV using spline interpolation is slower than VirCV and MinSearch, but still sufficiently fast for practical use in all tested cases and with the advantage of giving a *PRESS*-curve highly similar to the one obtained by the SegCV. The implicit segmented cross-validation (ImpCV) using Theorem B, is faster than SegCV for small segments and a bit slower for large segments, though still fast enough to provide exact results for all λ values. In general, the initial calculation of the SVD seems to be the main limiting factor in computational speed when the datasets grow in size. This is especially prominent for the milk data where SegCV performs this initial SVD 232 times. Here, a strategy avoiding SVD or using a randomised SVD algorithm [31] might be favourable, however, the other presented strategies are still usable.

V. DISCUSSION AND CONCLUSION

The essence of the TR-framework described in the present work is that just a single SVD-calculation (of either the original data matrix $\mathbf{X}$ or a transformed version $\tilde{\mathbf{X}}$) is required to explore some particular regularised regression problem of interest. We have pointed out that the *PRESS*- and *GCV* values required for model selection(s) based on the LooCV or the *GCV* can be obtained at the computational cost of two matrix-vector multiplications for each choice of the regularisation parameter value λ. In the applications section, it is demonstrated that our framework scales well when increasing the number of candidate regularisation parameter values in the case of 'small n with large p' problems. This scaling will also work well for problems involving multiple responses as most of the computations will be shared among responses. For smaller and medium-sized data as well as for other situations where the required SVD can be calculated (or approximated) reasonably fast, the acquired computational efficiency allows for the exploration of a large number of candidate models in a very short amount of time.

For situations where leave-one-out cross-validation underestimates validation error because of sample replicates or another grouping of samples, segmented cross-validation is the appropriate choice. We have proved a theorem saying that explicit remodelling for computation of cross-validated *PRESS*-values can be avoided, while still giving exact results, at the computational cost of inverting one matrix per sample segment per λ-value. For cases where the cost outweighs the benefits, we have proposed alternative strategies for reducing the number of inversions through careful selections of λ-values as well as an approximate virtual cross-validation (VirCV) strategy. The VirCV is a computationally efficient approximation of the traditional SegCV. In the applications (Section IV) we observed that the VirCV approximation of the SegCV appears to be quite accurate for model selection in the case of highly similar samples within each segment while using the LooCV or *GCV* in such situations is more likely to propose insufficient regularisation and models that predict poorer.

It is important to note that when the dataset is pre-processed and/or transformed by a data-dependent method, some bias both in the LooCV- and VirCV-based *PRESS* values must be expected. The data variable standardisation commonly used in RR is a typical example. The EMSC pre-processing that was used with or without replicate corrections is another. However, the main purpose of the LooCV- and VirCV-based *PRESS* values in the proposed framework is model selection rather than error estimation. The bias introduced by such pre-processing methods is therefore not likely to be very harmful as long as the (training) data does not contain serious outliers.

Although leverage correction of the model residuals for fast calculation of the LooCV in linear least squares regression problems is well known, there are some misleading assertions in the literature regarding both the properties and accuracy of *PRESS*-values that require clarification: i) Hansen [11, page 96] claims that the leverage values are not invariant under row permutations of the $\mathbf{X}$-data making the *PRESS*-values dependent on the ordering of the data. However, when the rows of the data matrix are permuted it can be verified that the leverage values are unchanged and undergo precisely the same permutation. Consequently, the correct leverage values will match up perfectly with the corresponding model residuals in the calculation of the *PRESS*(λ) calculations assuring its invariance under any row permutation of the $(\mathbf{X}, \mathbf{y})$-data. ii) Myers [32, page 399] claims that the expression for fast calculation of *PRESS*(λ) is only an approximation when performing centring and scaling of the data. This is, however, only true when the scaling factors are calculated from the data to be used in the model building. The data centring, as such, does not corrupt the leverage- and $PRESS(\lambda)$-values as long as the $1/n$ terms are included in the associated leverage corrections of the model residuals. iii) The version of Ridge regression implemented in the MASS package [33] for the R programming language includes a fast calculation of the $GCV(\lambda)$-values for a desired vector of corresponding λ-values. The $1/n$ term is, however, ignored when correcting the model residuals by the required averaged leverage value. Consequently, the resulting GCV-values are misleading when the centring of the data is included as a part of the Ridge Regression modelling.

We believe that future statistical texts and software dealing with Ridge Regression (and Tikhonov Regularisation) will find value in including the necessary pieces of linear algebra (in particular the simple matrix-vector multiplications of Equation (27) to establish the fast calculation of the $PRESS(\lambda)$ in Equation (26). In our opinion, these relatively simple but still powerful results demonstrate yet another remarkable consequence of the SVD at the core of applied multivariate data analysis.

Finally, we have established a theorem describing how to compute the cross-validated residuals for (regularised) linear regression models from the fitted value residuals. The computation can be seen as a multi-sample kind of leverage correction that applies to any type of segmented cross-validation strategy. In many cases, it represents a computationally efficient alternative to the computationally slower "hold out/remodelling approach" most common within statistics and machine learning. For the special case of LooCV, our theorem simplifies to the well-known scalar leverage correction calculations of the LooCV errors.

APPENDIX A
CALCULATING THE $\mathbf{B}_\lambda$-SOLUTIONS FROM THE SVD

The full SVD of $\mathbf{X} = \mathbf{USV}'$ yields $\mathbf{VV}' = \mathbf{I}_p$ and $\mathbf{X}'\mathbf{X} = \mathbf{VS}'\mathbf{SV}'$. The right singular vectors $\mathbf{V}$ of $\mathbf{X}$ are obviously eigenvectors for both $\mathbf{X}'\mathbf{X}$ and

$$\mathbf{X}_\lambda'\mathbf{X}_\lambda = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_p) = \mathbf{V}(\mathbf{S}'\mathbf{S} + \lambda\mathbf{I}_p)\mathbf{V}', \tag{34}$$

and their corresponding eigenvalues are given by the diagonals of $\mathbf{S}'\mathbf{S}$ and $\mathbf{S}'\mathbf{S} + \lambda\mathbf{I}_p$, respectively. The inverse matrix $(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I}_p)^{-1} = \mathbf{V}(\mathbf{S}'\mathbf{S} + \lambda\mathbf{I}_p)^{-1}\mathbf{V}'$, and the expression (10) for the TR-regression coefficients of a problem on standard form therefore simplifies [1] to

$$\begin{aligned}\mathbf{b}_\lambda &= \mathbf{V}(\mathbf{S}'\mathbf{S} + \lambda\mathbf{I}_p)^{-1}\mathbf{V}'\mathbf{VSU}'\mathbf{y}\\ &= \mathbf{V}(\mathbf{S}'\mathbf{S} + \lambda\mathbf{I}_p)^{-1}\mathbf{SU}'\mathbf{y}.\end{aligned} \tag{35}$$

In the following, we assume that $\mathbf{X}$ has full rank, i.e., $r = rank(\mathbf{X}) = \min(n, p)$. Then there are exactly r non-zero rows in the $\mathbf{S}$-factor of $\mathbf{b}_\lambda$, and the zero rows of $\mathbf{S}$ cancel both the associated columns in $\mathbf{V}(\mathbf{S}'\mathbf{S} + \lambda\mathbf{I}_p)^{-1}$ and rows in $\mathbf{U}'$. By considering the compact SVD of $\mathbf{X} = \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r'$ (the vanishing dimensions associated with the singular value 0 are omitted from the factorisation), the expression (35) for the regression coefficients $\mathbf{b}_\lambda$ simplifies to

$$\begin{aligned}\mathbf{b}_\lambda &= \mathbf{V}_r(\mathbf{S}_r^2 + \lambda\mathbf{I}_r)^{-1}\mathbf{S}_r\mathbf{U}_r'\mathbf{y}\\ &= \mathbf{V}_r(\mathbf{S}_r + \lambda\mathbf{S}_r^{-1})^{-1}\mathbf{U}_r'\mathbf{y} = \mathbf{V}_r\mathbf{c}_\lambda,\end{aligned} \tag{36}$$

where the coordinate vectors $\mathbf{c}_\lambda = (\mathbf{S}_r + \lambda\mathbf{S}_r^{-1})^{-1}\mathbf{U}_r'\mathbf{y} = [c_{\lambda,1}\ \dots\ c_{\lambda,r}]' \in \mathbb{R}^r$ has the scalar entries

$$c_{\lambda,j} = \frac{\mathbf{u}_j'\mathbf{y}}{s_j + \lambda/s_j}, \text{ for } 1 \leq j \leq r. \tag{37}$$

APPENDIX B
A FORMULA FOR THE SEGMENTED CROSS-VALIDATION RESIDUALS IN LINEAR LEAST SQUARES REGRESSION

The Sherman–Morrison–Woodbury updating formula for matrix inversion [14] says that

$$(\mathbf{A} + UCV)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{VA}^{-1}\mathbf{U})^{-1}\mathbf{VA}^{-1}, \tag{38}$$

where $\mathbf{A}$, $\mathbf{U}$, $\mathbf{C}$ and $\mathbf{V}$ are conformable matrices ($\mathbf{A}$ is $p \times p$, $\mathbf{C}$ is $k \times k$, $\mathbf{U}$ is $p \times k$, and $\mathbf{V}$ is $k \times p$). The matrix identity (38) means that the inverse of the rank-k modification of $\mathbf{A}$ on the left-hand side can be obtained from a rank-k modification of $\mathbf{A}^{-1}$ that includes inversion of two rank k matrices.

In the following we will use the notation

$$\begin{aligned}&\mathbf{A} = \mathbf{X}'\mathbf{X},\ \ \mathbf{U} = \mathbf{X}_{cv}',\ \ \mathbf{V} = \mathbf{X}_{cv},\\ &\mathbf{C} = -\mathbf{I}_k \text{ (the negative } k \times k \text{ identity matrix)},\end{aligned} \tag{39}$$

where the matrix $\mathbf{X}_{cv}$ denotes a cross-validation sample segment obtained by selecting some k rows from the full rank data matrix $\mathbf{X}$. Moreover, the vector $\mathbf{y}_{cv}$ denotes the corresponding selection of entries from the response vector $\mathbf{y}$. Finally, let $(\mathbf{X}_{(cv)}, \mathbf{y}_{(cv)})$ denote the remaining rows of the full dataset $(\mathbf{X}, \mathbf{y})$ that are not contained in the sample segment $(\mathbf{X}_{cv}, \mathbf{y}_{cv})$, where we assume that also $\mathbf{X}_{(cv)}$ has full rank.

Lemma

Let $\mathbf{M} = \mathbf{X}_{cv}(\mathbf{X}'\mathbf{X})^{-1}$ and $\mathbf{H}_{cv} = \mathbf{MX}_{cv}' = \mathbf{X}_{cv}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_{cv}'$. In the above notation, the following identity holds:

$$\mathbf{X}_{cv}(\mathbf{X}_{(cv)}'\mathbf{X}_{(cv)})^{-1} = [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{M}. \tag{40}$$

Proof:

By substitutions according to the identities from (39) into (38), we have:

$$\begin{aligned}&(\mathbf{X}_{(cv)}'\mathbf{X}_{(cv)})^{-1}\\ &= (\mathbf{X}'\mathbf{X} - \mathbf{X}_{cv}'\mathbf{X}_{cv})^{-1} = (\mathbf{X}'\mathbf{X} - \mathbf{X}_{cv}'\mathbf{I}_k\mathbf{X}_{cv})^{-1}\\ &= (\mathbf{X}'\mathbf{X})^{-1} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_{cv}'[\mathbf{I}_k - \mathbf{X}_{cv}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}_{cv}']^{-1}\\ &\quad\times \mathbf{X}_{cv}(\mathbf{X}'\mathbf{X})^{-1} = (\mathbf{X}'\mathbf{X})^{-1} + \mathbf{M}'[\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{M}.\end{aligned} \tag{41}$$

Multiplication of Equation (41) from the left by $\mathbf{X}_{cv}$ yields:

$$\begin{aligned}&\mathbf{X}_{cv}(\mathbf{X}_{(cv)}'\mathbf{X}_{(cv)})^{-1}\\ &= \mathbf{M} + \mathbf{H}_{cv}[\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{M}\end{aligned}$$

$$\begin{aligned} &= [\mathbf{I}_k - \mathbf{H}_{cv}][\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{M} + \mathbf{H}_{cv}[\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{M} \\ &= [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{M}. \qquad (42) \end{aligned}$$

■

By noting that

$$\mathbf{X}'_{(cv)}\mathbf{y}_{(cv)} = (\mathbf{X}'\mathbf{y} - \mathbf{X}'_{cv}\mathbf{y}_{cv}), \qquad (43)$$

we are in the position to prove the following result for the prediction residuals of segmented cross-validation (SegCV):

Theorem (SegCV):

The prediction residuals $\mathbf{r}_{(cv)} = \mathbf{y}_{cv} - \mathbf{X}_{cv}\hat{\boldsymbol{\beta}}_{(cv)}$ of the cross-validation sample segment $(\mathbf{X}_{cv}, \mathbf{y}_{cv})$ where $\hat{\boldsymbol{\beta}}_{(cv)} = (\mathbf{X}'_{(cv)}\mathbf{X}_{(cv)})^{-1}\mathbf{X}'_{(cv)}\mathbf{y}_{(cv)}$, can alternatively be obtained by a linear transformation of the associated fitted residuals $\mathbf{r}_{cv} = (\mathbf{y}_{cv} - \mathbf{X}_{cv}\hat{\boldsymbol{\beta}})$ from the full model $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ as follows:

$$\mathbf{r}_{(cv)} = [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{r}_{cv}. \qquad (44)$$

Proof:

$$\begin{aligned} &\mathbf{r}_{(cv)} \\ &= \mathbf{y}_{cv} - \mathbf{X}_{cv}\hat{\boldsymbol{\beta}}_{(cv)} \\ &= \mathbf{y}_{cv} - \underbrace{\mathbf{X}_{cv}(\mathbf{X}'_{(cv)}\mathbf{X}_{(cv)})^{-1}}_{(42)}\underbrace{\mathbf{X}'_{(cv)}\mathbf{y}_{(cv)}}_{(43)} \\ &= [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}[\mathbf{I}_k - \mathbf{H}_{cv}]\mathbf{y}_{cv} \\ &\quad - [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{M}[\mathbf{X}'\mathbf{y} - \mathbf{X}'_{cv}\mathbf{y}_{cv}] \\ &= [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}[\mathbf{y}_{cv} - \mathbf{H}_{cv}\mathbf{y}_{cv} - \mathbf{M}\mathbf{X}'\mathbf{y} + \mathbf{M}\mathbf{X}'_{cv}\mathbf{y}_{cv}] \\ &= [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}[\mathbf{y}_{cv} - \mathbf{H}_{cv}\mathbf{y}_{cv} \\ &\quad - \mathbf{X}_{cv}\underbrace{(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}}_{\hat{\boldsymbol{\beta}}} + \mathbf{H}_{cv}\mathbf{y}_{cv}] \\ &= [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}(\mathbf{y}_{cv} - \mathbf{X}_{cv}\hat{\boldsymbol{\beta}}) \\ &= [\mathbf{I}_k - \mathbf{H}_{cv}]^{-1}\mathbf{r}_{cv}. \qquad (45) \end{aligned}$$

■

Equation (44) shows that we can calculate the prediction residuals for a cross-validation sample segment of size k at the cost of inverting the $k \times k$ matrix $[\mathbf{I}_k - \mathbf{H}_{cv}]$ followed by a matrix-vector multiplication with the fitted residuals $\mathbf{r}_{cv}$.

The case of $k = 1$ corresponds to leave-one-out cross-validation (LooCV) where the prediction residual calculations reduce to scalar operations:

Corollary (LooCV):

The prediction residual $r_{(i)}$ when holding out the i-th sample $(\mathbf{x}_i, y_i)$ from the modelling is

$$r_{(i)} = r_i/(1 - h_i), \qquad (46)$$

where $r_i = y_i - \mathbf{x}_i\hat{\boldsymbol{\beta}}$ is the fitted residual and $h_i = \mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}'_i$. ■

Here, h_i is the i-th diagonal element of the projection matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ (projection onto the column space of $\mathbf{X}$).

APPENDIX C
THE TIKHONOV L_2-REGULARISATION FRAMEWORK

Tikhonov [10] noted that it is straightforward to generalise the above L_2 regularisation of $\mathbf{b}$ to more specialised types of regularisation through a corresponding regularisation matrix $\mathbf{L}$. These cases are expressed in terms of identifying the minimising solution of the bi-objective least squares problem

$$\begin{aligned} RSS_{\mathbf{L},\lambda}(\mathbf{b}) &= \|\mathbf{X}\mathbf{b} - \mathbf{y}\|^2 + \lambda\|\mathbf{L}\mathbf{b} - \mathbf{0}\|^2 \\ &= \|\mathbf{X}\mathbf{b} - \mathbf{y}\|^2 + \lambda\|\mathbf{L}\mathbf{b}\|^2, \qquad (47) \end{aligned}$$

for some fixed $\lambda > 0$. The minimisation of Equation (47) with respect to $\mathbf{b}$ can be obtained by considering the augmented data $\mathbf{X}_{\mathbf{L},\lambda} = \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{L} \end{bmatrix}$ and $\mathbf{y}_0 = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}$, and solving the normal equations

$$\mathbf{X}'_{\mathbf{L},\lambda}\mathbf{X}_{\mathbf{L},\lambda}\mathbf{b} = \mathbf{X}'_{\mathbf{L},\lambda}\mathbf{y}_0 \Rightarrow (\mathbf{X}'\mathbf{X} + \lambda\mathbf{L}'\mathbf{L})\mathbf{b} = \mathbf{X}'\mathbf{y} \qquad (48)$$

associated with the OLS problem $\mathbf{X}_{\mathbf{L},\lambda}\mathbf{b} = \mathbf{y}_0$.

To avoid technical distractions we will in the following restrict our attention to the cases of square and non-singular regularisation matrices $\mathbf{L}$ (even for situations where a non-square regularisation matrix is the immediate choice, a non-singular $(p \times p)$-alternative that provides a good approximation is often available). By defining $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{L}^{-1}$, the solution of the OLS problem in (48) is equivalent to finding the unique OLS-solution $\boldsymbol{\beta}_\lambda$ of the transformed problem $\tilde{\mathbf{X}}_\lambda\boldsymbol{\beta} = \mathbf{y}_0$, where $\tilde{\mathbf{X}}_\lambda = \mathbf{X}_{\mathbf{L},\lambda}\mathbf{L}^{-1} = \begin{bmatrix} \tilde{\mathbf{X}} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}$ and $\boldsymbol{\beta} = \mathbf{L}\mathbf{b}$. The associated expression minimised by $\boldsymbol{\beta}_\lambda$ is

$$\|\tilde{\mathbf{X}}\boldsymbol{\beta} - \mathbf{y}\|^2 + \lambda\|\boldsymbol{\beta}\|^2, \qquad (49)$$

i.e., in the standard form (5), and the minimising solution $\mathbf{b}_\lambda$ of the original problem (47) is obtained by

$$\mathbf{b}_\lambda = \mathbf{L}^{-1}\boldsymbol{\beta}_\lambda. \qquad (50)$$

Among the many useful choices for the regularisation matrix $\mathbf{L}$ are the following:

1) diagonal scaling (e.g., the standardisation of variables often advised for RR applications):

$$\mathbf{L}_{std} = \begin{bmatrix} \hat{\sigma}_1 & & & \\ & \hat{\sigma}_2 & & \\ & & \ddots & \\ & & & \hat{\sigma}_p \end{bmatrix},$$

where $\hat{\sigma}_i$ approximates the standard deviation of the i-th variable $(1 \leq i \leq p)$.

2) a (full) rank p discrete 1. derivative approximation:

$$\mathbf{L}_1 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ \sqrt{\epsilon}c_1 & \sqrt{\epsilon}c_1 & \ldots & \sqrt{\epsilon}c_1 & \sqrt{\epsilon}c_1 \end{bmatrix}.$$

3) a (full) rank p discrete 2. derivative approximation, as shown in the equation at the bottom of the page.

The alternatives $\mathbf{L}_1$ and $\mathbf{L}_2$ are relevant for problems where the $\mathbf{X}$-data are associated with discretised (uniform) sampling of continuous signals so that some smoothness in the solution candidates $\mathbf{b}_\lambda$ is reasonable. The two last rows in $\mathbf{L}_2$ (and the last row in $\mathbf{L}_1$) above are scaled versions of the discretised and normalised Legendre polynomials [34] of order 0 and 1, respectively (c_1 and c_2 represent the normalisation constants, and $\epsilon > 0$ is a scaling factor to be commented on below). It should be noted that both these row vectors are orthogonal to all the above rows in the derivative matrices where they appear.

The main purpose of the included Legendre vectors in these regularisation matrices is to ensure full rank of the regularisation matrices. Appropriate regularisation of the solutions $\mathbf{b}_\lambda$ may be obtained by choosing the fixed scaling factor $\epsilon > 0$ to be

- either sufficiently large to make $\mathbf{b}_\lambda$ practically orthogonal to the subspace spanned by the Legendre vectors, or
- sufficiently small to inhibit any notable penalisation with respect to the same Legendre vectors.

The choice of ϵ in the last case can therefore not be made arbitrarily small in practice. It must be chosen large enough to avoid numerical difficulties in the computations of $\tilde{\mathbf{X}}$ and $\mathbf{b}_\lambda$. Alternative (non-invertible) differentiation matrix candidates taking various boundary conditions into account are described in [11].

APPENDIX D
SPECIAL SITUATION FOR SEGMENT DECOMPOSITION IN VIRCV

In the following we will examine the proposed VirCV strategy more closely for three different situations:

a) Segments of identical rows.
b) Segments of collinear rows.
c) The general case (segments with no particular structure in the rows).

Identical Rows:

Let us assume that all the rows of a segment $\mathbf{X}_i$, $(1 \leq i \leq K)$ are identical. In this particular case, the *PRESS*-function associated with the VirCV is identical to the *PRESS*-function obtained by the SegCV.

The identity can be derived by noting that the left-multiplication of the left- and right-hand sides of a linear system by an orthogonal matrix affects neither the least squares solution nor the norm of the associated residual vector. Consequently, the SegCV strategy applied to the two systems (31) and (32) will result in identical *PRESS*-functions. With all rows within each segment $\mathbf{X}_k \in \mathbb{R}^{n_k \times p}$ being identical to its first row (denoted $\mathbf{x}_{k,1}$) of the segment, it is straightforward to verify that $\mathbf{X}_k$ has only one non-zero singular value $s_{k,1} = \sqrt{\mathbf{x}_{k,1}\mathbf{x}'_{k,1}n_k}$ and the corresponding left- and right singular vectors are

$$\mathbf{u}_{k,1} = \frac{1}{\sqrt{n_k}}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{n_k} \text{ and } \mathbf{v}_{k,1} = \frac{1}{\sqrt{\mathbf{x}_{k,1}\mathbf{x}'_{k,1}}}\mathbf{x}'_{k,1} \in \mathbb{R}^{p}. \tag{51}$$

By the orthogonality requirements of the SVD, any other left singular vector $\mathbf{u}$ must satisfy $\mathbf{u}'\mathbf{u}_{k,1} = 0$. Consequently

$$\mathbf{U}'_k\mathbf{X}_k = \begin{bmatrix} \sqrt{n_k}\mathbf{x}_{k,1} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \text{ and } \mathbf{U}'_k\mathbf{1} = \begin{bmatrix} \sqrt{n_k} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{52}$$

meaning that there will be only one non-zero row in each segment on the left-hand side of the $\tilde{\mathbf{T}}$-transformed system (32). It is therefore sufficient to demonstrate that the *PRESS*-functions obtained from applying the SegCV and the LooCV to the system in (32) are equal: Clearly, for any row containing just zeros in the left-hand side of (32) the prediction based on it is trivially identical to 0 (zero) for either of the cross-validation strategies (regardless of the regression coefficients). Because such zero rows do not contribute to the calculation of the regression coefficients, we are forced to conclude that the regression coefficients obtained by holding out the (only) non-zero row of a segment must be equal to the regression coefficients obtained from holding out the entire segment. Thus the predicted values for the non-zero row in each segment must also be identical for both cross-validation strategies, and we can conclude that the *PRESS* functions obtained by the SegCV- and the VirCV strategies must be identical.

Collinear (Proportional) Rows: One might expect the same result to hold when the rows within a segment are proportional. This is however not the case with the modelling strategy described above. The reason for this is that the

$$\mathbf{L}_2 = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ \sqrt{\epsilon}c_1 & \sqrt{\epsilon}c_1 & \dots & \sqrt{\epsilon}c_1 & \sqrt{\epsilon}c_1 & \sqrt{\epsilon}c_1 \\ -\sqrt{\epsilon}c_2\frac{p}{2} & -\sqrt{\epsilon}c_2\frac{p-1}{2} & \dots & \dots & \sqrt{\epsilon}c_2\frac{p-1}{2} & \sqrt{\epsilon}c_2\frac{p}{2} \end{bmatrix}.$$

inclusion of a constant term will make each of the K segments become a rank 2 – rather than a rank 1 submatrix. With more than one non-zero row on the left-hand side in each segment the argument of the previous situation fails, and doing LooCV on the transformed data is no longer equivalent to doing SegCV on the original data. However, when omitting the constant term from the modelling, each of the K segments has rank 1, and the SegCV and VirCV approaches will result in identical $PRESS(\lambda)$-functions. The rigorous explanation is similar to the argument given for the situation with identical rows.

APPENDIX E
TR PROTOTYPE MATLAB CODE

```
function [press, bcoefs, b, lambda, H, U, s, V, GCV, ...
          L, idmin, rescv] = TregsLooCV(X, y, lambdas, type)
% ----------------------------------------------------------
% INPUTS:
% X       - Data matrix
% y       - Response vector
% lambdas - Vector of regularisation parameter values
% type    - Regularisation type (-1 for standardisation,
%           0 for L2, 1 for 1st derivative regularisation,
%           etc ...)
% ----------------------------------------------------------
% OUTPUTS:
% press   - PRESS-statistic for input lambdas
% bcoefs  - Regression coefficients for selected lambda
%           (no constant term)
% b       - Regression coefficients for PRESS-minimal lambda
%           (with constant term)
% lambda  - Value of lambda minimising the PRESS-statistic
% H       - Vector of leverage values for all values of lambda
% U, s, V - SVD of matrix
% GCV     - GCV-statistic for input lambdas
% L       - Regularisation matrix
%           (empty for L2 regularisation)
% idmin   - Index of lambda value minimising
%           the PRESS-statistic
% rescv   - LooCV-residuals
% ----------------------------------------------------------

[n,p] = size(X);
mX = mean(X); my = mean(y);
X = bsxfun(@minus,X,mX); y = y-my;

L = [];
% Create full rank discrete derivative matrix of order 'type'.
if type > 0
    epsilon = 1e-14;
    L = diff([speye(p);sparse(type,p)],type);
    L(end-type+1:end,:) = sqrt(epsilon)*Plegendre(type-1,p);
elseif type < 0 % Create variable standardisation matrix.
    L = spdiags(std(X)',0,p,p);
end
if type ≠ 0, X = X/L; end

[U, S, V] = svd(X,'econ'); s = diag(S);
denom     = bsxfun(@plus,s,bsxfun(@rdivide,lambdas,s));
bcoefs    = V*bsxfun(@rdivide,(U'*y),denom);
H         = (U.^2)*bsxfun(@rdivide,s,denom)+1/n;
resid     = bsxfun(@minus,y, ...
                   U*bsxfun(@rdivide,s.*(U'*y),denom));
rescv     = bsxfun(@rdivide,resid,(1-H));
press     = sum(rescv.^2)';
GCV       = (sum(resid.^2)./mean(1-H).^2)';

% Finding press-minimal model and corresponding
% regression coefficients:
[¬,idmin] = min(press); lambda = lambdas(idmin);
h = H(:,idmin);
if type  ≠ 0, bcoefs = L\bcoefs; end
% Constant term
b         = [my-mX*bcoefs(:,idmin); bcoefs(:,idmin)];
end

function Q = Plegendre(d,p)
P = ones(p,d+1);
x = (-1:2/(p-1):1)';
for k = 1:d
    P(:,k+1) = x.^k;
end
[Q,¬] = qr(P,0);
Q = Q';
end
```

APPENDIX F
SEGCV PROTOTYPE MATLAB CODE

```
function [press, bcoefs, b, lambda, H, U, s, V, GCV, L, ...
          idmin, rescv] = TregsSegCV(X, y, lambdas, ...
                                     type, cv)
% ----------------------------------------------------------
% INPUTS:
% X       - Data matrix
% y       - Response vector
% lambdas - Vector of regularisation parameter values
% type    - Regularisation type (-1 for standardisation,
%           0 for L2, 1 for 1st derivative regularisation,
%           etc ...)
% cv      - Vector of cross-validation segments
%           (integers from 1 to #segments, length n)
% ----------------------------------------------------------
% OUTPUTS:
% press   - PRESS-statistic for input lambdas
% bcoefs  - Regression coefficients for selected lambda
%           (no constant term)
% b       - Regression coefficients for PRESS-minimal lambda
%           (with constant term)
% lambda  - Value of lambda minimising the PRESS-statistic
% H       - Vector of leverage values for all values of lambda
% U, s, V - SVD of matrix
% GCV     - GCV-statistic for input lambdas
% L       - Regularisation matrix
%           (empty for L2 regularisation)
% idmin   - Index of lambda value minimising the
%           PRESS-statistic
% rescv   - LooCV-residuals
% ----------------------------------------------------------

[n,p] = size(X);
nseg = max(cv); % Number of CV-segments
nlambda = length(lambdas);
mX = mean(X); my = mean(y);
X = bsxfun(@minus,X,mX); y = y-my;

L = [];
% Create full rank discrete derivative matrix of order 'type'.
if type > 0
    epsilon = 1e-14;
    L = diff([speye(p);sparse(type,p)],type);
    L(end-type+1:end,:) = sqrt(epsilon)*Plegendre(type-1,p);
elseif type < 0 % Create variable standardisation matrix.
    L = spdiags(std(X)',0,p,p);
end
if type ≠ 0, X = X/L; end

[U, S, V] = svd(X,'econ'); s = diag(S);
denom     = bsxfun(@plus,s,bsxfun(@rdivide,lambdas,s));
bcoefs    = V*bsxfun(@rdivide,(U'*y),denom);
H         = (U.^2)*bsxfun(@rdivide,s,denom)+1/n;
resid     = bsxfun(@minus,y,
                   U*bsxfun(@rdivide,s.*(U'*y),denom));
rescv     = zeros(n,nlambda);
sdenom  = sqrt(bsxfun(@rdivide,s,denom))';
for seg = 1:nseg
    Useg = U(cv==seg,:); I = eye(size(Useg,1)) - 1/n;
    for k = 1:nlambda
        Uk = Useg.*sdenom(k,:);
        % Exact CV using H correction
        rescv(cv==seg,k) = (I - Uk*Uk')\resid(cv==seg,k);
    end
end
press     = sum(rescv.^2)';
GCV       = (sum(resid.^2)./mean(1-H).^2)';

% Finding press-minimal model and corresponding
% regression coefficients:
[¬,idmin] = min(press); lambda = lambdas(idmin);
h = H(:,idmin);
if type  ≠ 0, bcoefs = L\bcoefs; end
% Constant term
b         = [my-mX*bcoefs(:,idmin); bcoefs(:,idmin)];
end
```

APPENDIX G
VIRCV PROTOTYPE MATLAB CODE

```
function [press, bcoefs, b, lambda, H, U, s, V, GCV, L, ...
         idmin, rescv, Usegments] = TregsVirCV(X, y, ...
                                   lambdas, type, segments)
% ----------------------------------------------------------
% INPUTS:
% X        - Data matrix
% y        - Response vector
% lambdas  - Vector of regularisation parameter values
% type     - Regularisation type (-1 for standardisation,
%            0 for L2, 1 for 1st derivative regularisation,
%            etc ...)
% segments   - List of integers identifying
%              cross-validation segments
% ---------------------------------------------------------
% OUTPUTS:
% press      - PRESS-statistic for input lambdas
% bcoefs     - Regression coefficients for selected lambda
%               (no constant term)
% b          - Regression coefficients for PRESS-minimal lambda
%               (with constant term)
% lambda     - Value of lambda minimising the PRESS-statistic
% H          - Vector of leverage values for all
%              values of lambda
% U, s, V    - SVD of matrix
% GCV        - GCV-statistic for input lambdas
% L          - Regularisation matrix
%               (empty for L2 regularisation)
% idmin      - Index of lambda value minimising the
%              PRESS-statistic
% rescv      - LooCV-residuals
% Usegments - Sparse matrix representing the orthogonal
%              transformations used in the VirCV
% ---------------------------------------------------------

% Finding orthogonal transformation and the modification
% to the leverage correction:
Usegments = segmentORTH(X, segments);
bs = (sum(Usegments,1).^2)';

[n,p] = size(X);
mX = mean(X); my = mean(y);
X = bsxfun(@minus,X,mX); y = y-my;

% Transforming data:
X = Usegments'*X; y = Usegments'*y;

L = [];
if type > 0
    epsilon = 1e-14;
    L = diff([speye(p);sparse(type,p)],type);
    P = Plegendre(type-1,p);
    L(end-type+1:end,:) = sqrt(epsilon)*P;
elseif type < 0
    L = spdiags(std(X)',0,p,p);
end

if type ≠ 0, X = X/L; end

[U, S, V]                 = svd(X,'econ'); s = diag(S);
s_plus_lambdas_over_s = bsxfun(@plus,s, ...
                                bsxfun(@rdivide,lambdas,s));

H      = bsxfun(@plus, (U.^2)*bsxfun(@ldivide,...
                s_plus_lambdas_over_s, s), bs/n);
bcoefs = V*bsxfun(@ldivide,s_plus_lambdas_over_s,(U'*y));
res    = bsxfun(@minus,y,X*bcoefs);
rescv  = bsxfun(@rdivide,res,(1-H));
press  = sum(rescv.^2)';
GCV    = sum(bsxfun(@rdivide,res,mean(1-H)).^2)';

if type ≠ 0, bcoefs = L\bcoefs; end

% Finding press-minimal model and corresponding
% regression coefficients:
[¬,idmin] = min(press); lambda = lambdas(idmin);
h = H(:,idmin);
if type  ≠ 0, bcoefs = L\bcoefs; end
% Constant term
b         = [my-mX*bcoefs(:,idmin); bcoefs(:,idmin)];

end

function U = segmentORTH(X, segments)
n         = size(X,1);
nsegments = max(segments);
U         = sparse(n,n);
for k = 1:nsegments
    ind              = find(segments==k);
    [U(ind, ind),¬] = svd(X(ind,:),'econ');
end
end

function Q = Plegendre(d,p)
P = ones(p,d+1);
x = (-1:2/(p-1):1)';
for k = 1:d
    P(:,k+1) = x.^k;
end
[Q,¬] = qr(P,0);
Q = Q';
end
```

APPENDIX H
COMPUTATIONAL COMPLEXITY OF THE FAST LOOCV

For a more precise description of the computational complexity involved in calculating the fast LooCV, an approximate count of the floating point operations (flop) is required. According to Björck [17], an approximate flop count for finding the reduced SVD (using a QR-SVD algorithm with Golub–Kahan–Householder bidiagonalisation) of a $(n \times p)$-matrix is $12pn^2 + (16/3)n^3$ when assuming $p \geq n$. The remaining computations consist of centring, calculating the *PRESS* values, and calculating the *PRESS*-minimal regression coefficients for every response. With q different responses, the approximate flop count for these computations is given by:

$$\begin{aligned}(3np + 3nq + nr + 2nrq - q + 2prq + pq)\\ + n_\lambda(3r + 2nr + 2nrq + qr + 4nq),\end{aligned} \tag{53}$$

where n_λ denotes the number of different candidate regularisation parameter values. For $p \geq n$, the computations needed to evaluate the $PRESS(\lambda)$-function for one additional regularisation parameter is of the order $\mathcal{O}(qn^2)$, and in particular the additional computations are independent of the number (p) of measured features. This makes the fast LooCV highly useful also for problems where the number of features is even larger than the number of samples. To calculate the cost of finding the corresponding $GCV(\lambda)$-values as well as GCV-minimal regression coefficients one should add $5nn_\lambda q - q + q(2pr + p)$ to the above flop count. Note that the choice of regularisation matrix $\mathbf{L}$ matters here, and for $\mathbf{L} \neq \mathbf{I}$ there are additional calculations (see Section C) that must be taken into account. The exact number of flops associated with these additional calculations will depend on the sparsity structure of $\mathbf{L}$ and to what extent that sparsity can be utilised in the required calculations.

TABLE 8. Approximate flop counts for the required SVD(s) in the different parameter selection methods when assuming $p \geq n$.

Par. sel. method	Approx. flops for SVD(s)
LooCV/GCV	$12pn^2 + \frac{16}{3} \cdot n^3$
VirCV	$12pn^2 + \frac{16}{3} \cdot n^3 + K \cdot \left(12p \cdot B_{ss}^2 + \frac{16}{3} \cdot B_{ss}^3\right)$
SegCV	$K \cdot \left(12p \cdot (n - B_{ss})^2 + \frac{16}{3} \cdot (n - B_{ss})^3\right)$

APPENDIX I
COMPUTATIONAL SAVINGS OF THE VIRCV COMPARED TO THE SEGCV

To assess the computational savings of the VirCV over the SegCV, flop count approximations for the associated *PRESS*-values must be compared. (We only consider the situation involving L_2 regularisation, i.e. the identity matrix $\mathbf{I}$ acting as the regularisation matrix.) Let K denote the number of segments, and assume for simplicity that the various segment sizes are all bounded from above by the constant B_{ss}. The approximate number of flops required for the SVDs for the different parameter selection methods when using the entire dataset for training are given by the formulas in Table 8 (using the approximate flop count for the SVD given in [17]). The Table shows that the size of (all but one of) the required SVDs for the VirCV are much smaller than for the SegCV (assuming the size of each segment is much smaller than the total number of samples, which is obviously the case in most real applications). This is primarily what makes the

VirCV superior to the SegCV in terms of computational efficiency.

If the block diagonal structure of the transformation matrix T is utilised, the matrix multiplication part of the orthogonal transformation (32) for the VirCV requires approximately

$$2B_{ss}(B_{ss}-1)+K\cdot B_{ss}\cdot p(2B_{ss}-1)+q\cdot B_{ss}(2B_{ss}-1) \tag{54}$$

flops. For keeping track of the remaining computations needed for the VirCV we can use the flop count approximations in Section H, as the flop count for the VirCV and the LooCV will be identical after applying the orthogonal transformation required for the VirCV. The approximate flop count of the remaining computations for the SegCV is given by

$$\begin{aligned}2K\cdot B_{ss}(q+p)-q+r_{train}\cdot q(2B_{ss}-1)\\ +q\cdot n_{\lambda}\cdot K[3r_{train}+2p\cdot r_{train}+p+2p\cdot n_{test}+3n_{test}]\end{aligned} \tag{55}$$

where $r_{train} = \min(n_{train}, p)$ and n_{train} is the number of samples in the training set.

Although the main computational cost with model validation is with the initial SVD(s) there will also be an additional computational cost for each candidate regularisation parameter value for which we want to validate the model. Consider the case $p > n$ of most interest for the present work (the number of features is greater than the number of samples). From the above reasoning, we observe that when considering additional regularisation parameter values, the SegCV flop count depends on the number of features p for each candidate value. The above flop count for the VirCV and the LooCV flop count in Appendix H shows that this is not the case for the VirCV. When p is very large it might therefore be computationally inefficient (or even infeasible) to validate models for a large number of regularisation parameter values based on the SegCV. Clearly, the VirCV is the method of choice among the two in such cases.

REFERENCES

[1] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Series in Statistics Springer), vol. 1. Berlin, Germany: Springer, 2009.

[2] J. S. U. Hjorth, *Computer Intensive Statistical Methods: Validation, Model Selection and Bootstrap*. Boca Raton, FL, USA: CRC Press, 1993.

[3] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Stat. Soc., Ser. B*, vol. 36, no. 2, pp. 111–147, 1974.

[4] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, May 1979.

[5] D. Benkeser, M. Petersen, and M. J. van der Laan, "Improved small-sample estimation of nonlinear cross-validated prediction metrics," *J. Amer. Stat. Assoc.*, vol. 115, no. 532, pp. 1917–1932, Oct. 2020.

[6] S. Bates, T. Hastie, and R. Tibshirani, "Cross-validation: What does it estimate and how well does it do it?" *J. Amer. Stat. Assoc.*, pp. 1–12, Mar. 2023, doi: 10.1080/01621459.2023.2197686.

[7] G. C. S. Smith, S. R. Seaman, A. M. Wood, P. Royston, and I. R. White, "Correcting for optimistic prediction in small data sets," *Amer. J. Epidemiol.*, vol. 180, no. 3, pp. 318–324, Aug. 2014.

[8] A. Celisse and B. Guedj, "Stability revisited: New generalisation bounds for the leave-one-out," 2016, *arXiv:1608.06412*.

[9] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970.

[10] A. N. Tikhonov, "On the solution of ill-posed problems and the method of regularization," *Doklady Akademii Nauk*, vol. 151, no. 3, pp. 501–504, 1963.

[11] P. C. Hansen, *Discrete Inverse Problems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2010, doi: 10.1137/1.9780898718836.

[12] D. M. Allen, "Mean square error of prediction as a criterion for selecting variables," *Technometrics*, vol. 13, no. 3, pp. 469–475, Aug. 1971.

[13] D. M. Allen, "The relationship between variable selection and data agumentation and a method for prediction," *Technometrics*, vol. 16, no. 1, pp. 125–127, Feb. 1974.

[14] A. S. Householder, *The Theory of Matrices in Numerical Analysis* (Blaisdell Book in the Pure and Applied Sciences). Waltham, MA, USA: Blaisdell, 1965.

[15] J. H. Kalivas, "Overview of two-norm (L_2) and one-norm (L_1) Tikhonov regularization variants for full wavelength or sparse spectral multivariate calibration models or maintenance," *J. Chemometrics*, vol. 26, no. 6, pp. 218–230, Jun. 2012.

[16] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *J. ACM*, vol. 9, no. 1, pp. 84–97, Jan. 1962.

[17] Å. Björck, *Numerical Methods in Matrix Computations*. Berlin, Germany: Springer, 2016.

[18] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1973, ch. 4.

[19] U. Indahl, "A twist to partial least squares regression," *J. Chemometrics*, vol. 19, no. 1, pp. 32–44, Jan. 2005.

[20] J. H. Kalivas, "Two data sets of near infrared spectra," *Chemometric Intell. Lab. Syst.*, vol. 37, no. 2, pp. 255–259, Jun. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169743997000385

[21] L. B. Lyndgaard, K. M. Sørensen, F. van den Berg, and S. B. Engelsen, "Depth profiling of porcine adipose tissue by Raman spectroscopy," *J. Raman Spectrosc.*, vol. 43, no. 4, pp. 482–489, Apr. 2012.

[22] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers, "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, Mar. 2002.

[23] C. D. Brown and R. L. Green, "Critical factors limiting the interpretation of regression vectors in multivariate calibration," *TrAC Trends Anal. Chem.*, vol. 28, no. 4, pp. 506–514, Apr. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165993609000363

[24] N. K. Afseth, J. P. Wold, and V. H. Segtnan, "The potential of Raman spectroscopy for characterisation of the fatty acid unsaturation of salmon," *Analytica Chim. Acta*, vol. 572, no. 1, pp. 85–92, Jul. 2006.

[25] N. K. Afseth and A. Kohler, "Extended multiplicative signal correction in vibrational spectroscopy, a tutorial," *Chemometrics Intell. Lab. Syst.*, vol. 117, pp. 92–99, Aug. 2012.

[26] K. A. Kristoffersen, K. H. Liland, U. Böcker, S. G. Wubshet, D. Lindberg, S. J. Horn, and N. K. Afseth, "FTIR-based hierarchical modeling for prediction of average molecular weights of protein hydrolysates," *Talanta*, vol. 205, Dec. 2019, Art. no. 120084.

[27] N. K. Afseth, H. Martens, Å. Randby, L. Gidskehaug, B. Narum, K. Jørgensen, S. Lien, and A. Kohler, "Predicting the fatty acid composition of milk: A comparison of two Fourier transform infrared sampling techniques," *Appl. Spectrosc.*, vol. 64, no. 7, pp. 700–707, Jul. 2010.

[28] Å. T. Randby, M. R. Weisbjerg, P. Nørgaard, and B. Heringstad, "Early lactation feed intake and milk yield responses of dairy cows offered grass silages harvested at early maturity stages," *J. Dairy Sci.*, vol. 95, no. 1, pp. 304–317, Jan. 2012.

[29] K. H. Liland, A. Kohler, and N. K. Afseth, "Model-based pre-processing in Raman spectroscopy of biological samples," *J. Raman Spectrosc.*, vol. 47, no. 6, pp. 643–650, Jun. 2016.

[30] A. Kohler, U. Böcker, J. Warringer, A. Blomberg, S. W. Omholt, E. Stark, and H. Martens, "Reducing inter-replicate variation in Fourier transform infrared spectroscopy by extended multiplicative signal correction," *Appl. Spectrosc.*, vol. 63, no. 3, pp. 296–305, Mar. 2009.

[31] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, Jan. 2011.

[32] R. H. Myers, *Classical and Modern Regression With Applications*, vol. 1. Boston, MA, USA: PWS-Kent, 1990.

[33] W. N. Venables and B. D. Ripley, *Modern Applied Statistics With S*, 4th ed. New York, NY, USA: Springer, 2002. [Online]. Available: https://cran.r-project.org/web/packages/MASS/index.html, and http://www.stats.ox.ac.uk/pub/MASS4

[34] E. Kreyszig, *Introductory Functional Analysis With Applications*, vol. 1. New York, NY, USA: Wiley, 1978.

KRISTIAN HOVDE LILAND received the Ph.D. degree in applied statistics, in 2010. He is currently a Professor of statistics with the Faculty of Science and Technology, Norwegian University of Life Sciences. His research interests include multivariate statistics and machine learning with a large spectrum of projects ranging from theoretic to applied and often, with an emphasis on scientific programming.

JOAKIM SKOGHOLT received the M.Sc. degree in mathematics from the University of Warwick, in 2012, and the Ph.D. degree in applied mathematics and data analysis from the Norwegian University of Life Sciences (NMBU), in 2019. He is currently an Assistant Professor with NMBU. His research interests include machine learning and multivariate statistics.

ULF GEIR INDAHL received the Ph.D. degree in applied mathematics from the University of Oslo, in 1998. He is currently a Professor of statistics with the Faculty of Science and Technology, Norwegian University of Life Sciences. His research interests include the development of new methodologies and applications ranging from high-dimensional multivariate data analysis to machine learning and deep learning.