Lukas, M.A., de Hoog, F.R. and Anderssen, R.S. (2010) Efficient algorithms for robust generalized cross-validation spline smoothing. Journal of Computational and Applied Mathematics, 235 (1). pp. 102-107.

Accepted Manuscript

Efficient algorithms for robust GCV spline smoothing

Mark A. Lukas, Frank R. de Hoog, Robert S. Anderssen

PII: S0377-0427(10)00292-X
DOI: 10.1016/j.cam.2010.05.016
Reference: CAM 7910

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 23 July 2009
Revised date: 23 April 2010

Please cite this article as: M.A. Lukas, F.R. de Hoog, R.S. Anderssen, Efficient algorithms for robust GCV spline smoothing, *Journal of Computational and Applied Mathematics* (2010), doi:10.1016/j.cam.2010.05.016

Efficient algorithms for robust GCV spline smoothing

Mark A. Lukas[a,*], Frank R. de Hoog[b], Robert S. Anderssen[b]

[a] *Mathematics and Statistics, Murdoch University, South Street, Murdoch WA 6150, Australia*
[b] *CSIRO Mathematics, Informatics and Statistics, GPO Box 664, Canberra ACT 2601, Australia*

Abstract

Generalized cross-validation (GCV) is a widely used parameter selection criterion for spline smoothing, but it can give poor results if the sample size n is not sufficiently large. An effective way to overcome this is to use the more stable criterion called robust GCV (RGCV). The main computational effort for the evaluation of the GCV score is the trace of the smoothing matrix, $\mathrm{tr}A$, while the RGCV score requires both $\mathrm{tr}A$ and $\mathrm{tr}A^2$. Since 1985 there has been an efficient $O(n)$ algorithm to compute $\mathrm{tr}A$. This paper develops two pairs of new $O(n)$ algorithms to compute $\mathrm{tr}A$ and $\mathrm{tr}A^2$, which allow the RGCV score to be calculated efficiently. The algorithms involve the differentiation of certain matrix functionals using banded Cholesky decomposition.

Keywords: Cholesky decomposition, Generalized cross-validation, Smoothing matrix, Smoothing parameter, Spline, Trace
2000 MSC: 65F30, 65D10, 62G08

1. Introduction

A common task in data analysis is to fit a smooth curve to noisy data

$$y_i = f(x_i) + \varepsilon_i, \quad a \le x_1 < x_2 < \cdots < x_n \le b, \quad i = 1, \ldots, n, \qquad (1)$$

*Corresponding author.
Email addresses: M.Lukas@murdoch.edu.au (Mark A. Lukas), Frank.deHoog@csiro.au (Frank R. de Hoog), Bob.Anderssen@csiro.au (Robert S. Anderssen)

where it is assumed that the random errors ε_i are uncorrelated with zero mean and equal variance σ^2. Smoothing splines are widely used for this purpose [1, 2]. The natural polynomial smoothing spline of degree $2m-1$ can be defined as the minimizer f_λ of the functional

$$n^{-1}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\int_a^b (f^{(m)}(x))^2 dx \tag{2}$$

over all functions f for which $f^{(m)}$ is square integrable. The most frequently used smoothing spline is the cubic spline, for which $m = 2$.

It is well known that the quality of the fit depends critically on the choice of the smoothing parameter $\lambda > 0$. If λ is too small, then f_λ is too rough, and, if λ is too large, then f_λ is overly smooth and is not faithful to the data. In fact, as $\lambda \to \infty$, f_λ approaches the least squares polynomial of degree $m-1$. A popular and practical parameter selection criterion is generalized cross-validation (GCV) [3, 2]. This criterion usually performs well for problems with large sample size n. However, it can be unreliable for smaller n, and, even for large n, it occasionally gives a parameter value that is far too small. For this reason, two more stable extensions of GCV called robust GCV (RGCV) [4, 5, 6] and modified GCV [7] were developed. The RGCV and modified GCV criteria have favourable small-sample and large-sample properties, and they perform very well in simulations [8, 9].

Denote $\boldsymbol{f}_\lambda = (f_\lambda(x_1), \ldots, f_\lambda(x_n))^T$ and let $A(\lambda)$ be the smoothing matrix defined by $\boldsymbol{f}_\lambda = A(\lambda)\boldsymbol{y}$. The main computational effort in using GCV is the calculation of the trace $\mathrm{tr}A(\lambda)$, which is often referred to as the degrees of freedom for the spline. With the representation of f_λ in [10, 11], the smoothing matrix can be written in terms of the inverse of a certain banded matrix of bandwidth $2m+1$. Using the band structure and Cholesky decomposition, Hutchinson and de Hoog [12] developed an efficient $O(m^2n)$ algorithm to compute the diagonal elements of $A(\lambda)$, and hence to find $\mathrm{trA}(\lambda)$. The diagonal elements of $A(\lambda)$, called the leverage values, are also used to obtain confidence intervals for the spline estimate [13]. With the local support basis for f_λ in [14], the method in [15] yields another $O(m^2n)$ algorithm for the calculation of $\mathrm{trA}(\lambda)$ and the leverage values (see also [16, sect. 3.8.1]). There are also efficient $O(m^2n)$ algorithms for the GCV criterion based on QR factorization [17, 18]. The modified GCV criterion requires the same calculations as GCV, and so the same $O(m^2n)$ algorithms can be used.

The aim of this paper is to develop and investigate efficient exact algorithms for the RGCV criterion. The RGCV score function requires the cal-

culation of both $\mathrm{tr}A(\lambda)$ and $\mathrm{tr}A^2(\lambda)$. We develop two pairs of new $O(m^2n)$ algorithms to calculate these quantities. The algorithms use an approach involving the differentiation of certain matrix functionals [19], and are based on the Cholesky decomposition of a banded matrix. One of the algorithms for $\mathrm{tr}A(\lambda)$ is similar to the algorithm in [12].

In addition to exact methods, there are other methods that approximate $\mathrm{tr}A(\lambda)$ and $\mathrm{tr}A^2(\lambda)$. In particular, using the known asymptotic behaviour of the eigenvalues τ_i in the Demmler-Reinsch diagonalization $A(\lambda) = Q\mathrm{diag}(1+\lambda\tau_i)^{-1}Q^T$, asymptotic estimates of both $\mathrm{tr}A(\lambda)$ and $\mathrm{tr}A^2(\lambda)$ can be obtained [20]. The estimate of $\mathrm{tr}A(\lambda)$ was used in [21] to derive an asymptotic GCV selection criterion. By using the asymptotic estimate of $\mathrm{tr}A^2(\lambda)$, one can also derive an asymptotic RGCV criterion. A different approach is to use a stochastic estimator of $\mathrm{tr}A(\lambda)$ and so approximate the GCV score [22, 23]. This can be extended easily to estimate $\mathrm{tr}A^2(\lambda)$ with little extra effort (since, having estimated $\mathrm{tr}A(\lambda)$ by $\boldsymbol{u}^TA\boldsymbol{u}$ for a pseudo-random vector $\boldsymbol{u}$, then $\mathrm{tr}A^2(\lambda)$ can be estimated by $\|A\boldsymbol{u}\|^2$).

Besides its use in the RGCV criterion, the function $\mathrm{tr}A^2(\lambda)$ also arises in the variance estimate [24]

$$\hat{\sigma}^2 = \|(I - A(\lambda))\boldsymbol{y}\|^2/\mathrm{tr}((I - A(\lambda))^2),$$

where $A(\lambda)$ is the smoothing matrix above. Therefore, the algorithms developed here also apply to the calculation of $\hat{\sigma}^2$. The three quantities $\mathrm{tr}A$, $\mathrm{tr}A^2$ and $\mathrm{tr}(2A - A^2)$, which coincide for parametric linear regression, all have useful interpretations as degrees of freedom in the smoothing situation [25].

The paper is organized as follows. After some preliminaries in Section 2, the first pair of algorithms for $\mathrm{tr}A(\lambda)$ and $\mathrm{tr}A^2(\lambda)$ is developed in Section 3. These algorithms use an approach that yields the diagonals of $A(\lambda)$ and $A^2(\lambda)$. The second pair of algorithms, based on log determinant relationships for $\mathrm{tr}A(\lambda)$ and $\mathrm{tr}A^2(\lambda)$, is developed in Section 4. We compare the efficiencies of the algorithms in Section 5.

2. Preliminaries

The smoothing spline f_λ can be computed efficiently using a local support spline basis for $f_\lambda^{(m)}$ [26, 10, 11], that is, with the representation

$$f_\lambda^{(m)} = \sum_{i=1}^{n-m} c_i M_i,$$

where the M_i are B-splines. Owing to the continuity conditions at the knots, the spline f_λ is uniquely determined by the coefficients $\boldsymbol{c} = (c_1, \ldots, c_{n-m})^T$ and values $\boldsymbol{a} = (f_\lambda(x_1), \ldots, f_\lambda(x_n))^T$. These coefficients and values can be computed by solving

$$(H + n\lambda G^T G)\boldsymbol{c} = G^T \boldsymbol{y}, \tag{3}$$

$$\boldsymbol{a} = \boldsymbol{y} - n\lambda G\boldsymbol{c}, \tag{4}$$

where H and $G^T G$ are symmetric, positive definite band matrices of bandwidth $2m-1$ and $2m+1$, respectively. The $(n-m) \times (n-m)$ matrix H has elements

$$h_{ij} = \int_a^b M_i(x) M_j(x) dx$$

and the $n \times (n-m)$ matrix G is an (m+1)-banded lower triangular matrix with the non-zero elements of column i equal to the coefficients of the mth divided difference based on $x_i, \ldots, x_{i+m}$.

Let $p = \lambda^{-1}$ and define $B = B(p) = nG^T G + pH$. Then, from (3) and (4), the smoothing matrix $A(\lambda)$, defined by $\boldsymbol{f}_\lambda = A(\lambda)\boldsymbol{y}$, satisfies

$$I - A(\lambda) = nGB^{-1}(p)G^T. \tag{5}$$

Note that $A(\lambda)$ is symmetric and $I - A(\lambda)$ is non-negative definite.

The GCV criterion selects λ as the minimizer of the GCV function

$$V(\lambda) = \frac{n^{-1}\|(I - A(\lambda))\boldsymbol{y}\|^2}{[n^{-1}\mathrm{tr}(I - A(\lambda))]^2}, \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm. Let $\mu_2(\lambda)$ denote the normalized trace $\mu_2(\lambda) = n^{-1}\mathrm{tr}A^2(\lambda)$. The RGCV criterion selects λ as the minimizer of the weighted sum

$$\overline{V}(\lambda) = [\gamma + (1-\gamma)\mu_2(\lambda)]V(\lambda), \tag{7}$$

where $\gamma \in (0, 1)$ is a robustness parameter. Clearly, the RGCV function requires the computation of $\|(I - A(\lambda))\boldsymbol{y}\|^2$ and $\mathrm{tr}(I - A(\lambda))$ (as in GCV) and also $\mathrm{tr}A^2(\lambda)$.

Obviously, when $\gamma = 1$, the RGCV criterion reduces to GCV. As γ is decreased from 1, the RGCV criterion becomes more stable and it performs very well for $\gamma \in [0.2, 0.4]$ [9]. Note that for both GCV and RGCV, no knowledge of the error variance is required.

The function $\mu_2(\lambda)$ is important in its own right. The reason for this, and why it is in (7), is that it is proportional to the variance $v(\lambda)$ of the

spline in a mean square sense. In fact, under the assumption of uncorrelated errors with equal variance σ^2, the variance equals

$$v(\lambda) = n^{-1}E\|\boldsymbol{f}_\lambda - E\boldsymbol{f}_\lambda\|^2 = \sigma^2 n^{-1}\mathrm{tr}(A^T(\lambda)A(\lambda)) = \sigma^2\mu_2(\lambda),$$

where E denotes expectation.

To minimize the GCV and RGCV scores in practice, the functions $V(\lambda)$ and $\overline{V}(\lambda)$ need to be computed efficiently for many different values of λ. Because B is banded, symmetric and positive definite, one can efficiently compute the Cholesky decomposition $B = U^TU$, where $U = U(p)$ is an $(m+1)$-banded upper triangular matrix with positive diagonal elements. This requires approximately $(m^2+3m)n/2$ operations (assuming $m << n$) and $(n-m)$ square roots [27]. Here and throughout, the approximate operation count has the correct $O(n)$ term, and an operation consists of a multiplication/division and an addition/subtraction. As in [10], the sum of squared residuals is

$$\|(I - A(\lambda))\boldsymbol{y}\|^2 = \|nGB^{-1}(p)G^T\boldsymbol{y}\|^2 = n^2\|G\boldsymbol{z}\|^2, \tag{8}$$

where $\boldsymbol{z}$ satisfies $U^TU\boldsymbol{z} = G^T\boldsymbol{y}$. This can be used to efficiently compute the numerator of $V(\lambda)$ in approximately $(4(m+1)+1)n$ additional operations. Similarly, the numerator can be computed in the same number of operations from a Cholesky decomposition of $H + n\lambda G^TG$.

The next two sections present algorithms for the computation of $\mathrm{tr}(I - A(\lambda))$ and $\mathrm{tr}A^2(\lambda)$. Because these algorithms are also based on a Cholesky decomposition, we will suppose, for the operation counts, that this decomposition is already known.

3. Algorithms that compute the diagonals

In this section, we develop algorithms for $\mathrm{tr}(I - A(\lambda))$ and $\mathrm{tr}A^2(\lambda)$ using an approach that yields the diagonals of $A(\lambda)$ and $A^2(\lambda)$. In the first algorithm, $\mathrm{tr}(I - A(\lambda))$ is calculated in a similar way to that in [12].

From the Cholesky decomposition $B = U^TU$, let $S = \mathrm{diag}(u_{ii}^{-1})$. In addition, denote $B^{-1} = [\hat{b}_{ij}]$. The equation $B = U^TU$ can be written in the form

$$B^{-1} = SU^{-T} + (I - SU)B^{-1},$$

where SU^{-T} is lower triangular with diagonal $[s_{11}^2, \ldots, s_{n-m,n-m}^2]$ and SU is unit upper triangular. By considering the diagonal part and the strictly

upper triangular part of this equation, the elements $\hat{b}_{ij}$ of the upper central $m+1$ band of B^{-1} can be computed recursively as in [12], starting with $\hat{b}_{n-m,n-m} = u_{n-m,n-m}$. This requires approximately $((m+1)^2+2)n$ operations (approximately $2n$ for S^2, $(m+1)n$ for SU, and $m(m+1)n$ for the recursive procedure in [12]), and, since B^{-1} is symmetric, it gives the central $2m+1$ band of B^{-1}. With this band, we can compute the diagonal elements of G^TGB^{-1} in approximately $(m+1)n$ operations (utilizing the symmetry of G^TG and B^{-1}), and thereby find

$$\mathrm{tr}(I - A(\lambda)) = \mathrm{tr}(nGB^{-1}(p)G^T) = \mathrm{tr}(nG^TGB^{-1}(p)),$$

where the last equality follows from the general identity $\mathrm{tr}(XY) = \mathrm{tr}(YX)$.

Alternatively, with the central $2m+1$ band of B^{-1}, we can compute the diagonal elements of $nGB^{-1}(p)G^T$ in approximately $(m+1)(m+2)n$ operations ($(m+1)^2n$ operations for GB^{-1} and $(m+1)n$ operations for the diagonal of $GB^{-1}G^T$) and thereby find $\mathrm{tr}(I - A(\lambda))$. This also yields the diagonal elements of $A(\lambda)$, the leverage values, which are useful for construction of confidence intervals for the spline estimate [13]. Thus, we can compute $\mathrm{tr}(I - A(\lambda))$ in about $((m+1)^2 + m + 3)n$ operations without the leverage values or about $(2(m+1)^2 + m + 3)n$ operations with the leverage values.

Remark. The algorithm in [12], which is based on a rational Cholesky decomposition of B, requires approximately $(m+1)^2n$ operations without leverage values and approximately $2(m+1)^2n$ operations with leverage values, and so it is somewhat more efficient than the algorithm above. However, for our approach, the ordinary Cholesky decomposition above is better suited to the calculation of $\mathrm{tr}A^2(\lambda)$.

To compute $\mathrm{tr}A^2(\lambda)$, first note that, from (5),

$$\begin{aligned}\mathrm{tr}A^2 &= n - 2\mathrm{tr}(I - A) + \mathrm{tr}(I - A)^2 \\ &= m + \mathrm{tr}(I - nB^{-1}G^TG)^2 = m + p^2\mathrm{tr}(B^{-1}HB^{-1}H). \end{aligned} \tag{9}$$

The right hand side of (9) can be computed using the fact that

$$\frac{dB^{-1}}{dp} = -B^{-1}\frac{dB}{dp}B^{-1} = -B^{-1}HB^{-1}, \tag{10}$$

where the first equality follows from differentiating $BB^{-1} = I$ [28, p. 307].

To compute dB^{-1}/dp, we first compute the derivative $U' = dU/dp$. Differentiation of $U^TU = B(p)$ yields

$$(U')^TU + U^TU' = dB/dp = H.$$

This equation can be solved recursively for the elements of U', starting with the first row of U'. The procedure is:
For $i = 1, \ldots, n-m$

$$u'_{ii} = \left(\frac{h_{ii}}{2} - \sum_{k=\max\{1,i-m\}}^{i-1} u'_{ki} u_{ki} \right) / u_{ii} \tag{11}$$

For $j = i+1, \ldots, \min\{i+m, n-m\}$

$$u'_{ij} = \left(h_{ij} - u'_{ii} u_{ij} - \sum_{k=\max\{1,j-m\}}^{i-1} (u'_{ki} u_{kj} + u_{ki} u'_{kj}) \right) / u_{ii} \tag{12}$$

The multiplications/divisions involved in this procedure are in a one-to-one correspondence with those needed to compute $(U')^T U + U^T U' = (U^T U')^T + U^T U'$, i.e. to compute $U^T U'$, for known (m+1)-banded matrices U and U'. Therefore, the number of operations required is approximately $(m+1)^2 n$.

To obtain a system for $(B^{-1})' = dB^{-1}/dp$, we differentiate the equation $B^{-1} U^T = U^{-1}$ to obtain

$$(B^{-1})' U^T + B^{-1} U'^T = -U^{-1} U' U^{-1}.$$

By considering the lower triangular part of this equation, it follows that

$$((B^{-1})' U^T)_{ij} = -(B^{-1} U'^T)_{ij} - u_{ii}^{-2} u'_{ij} \delta_{ij}, \quad i \geq j.$$

From these equations, the elements $\hat{b}'_{ij}$ in the central $2m+1$ band of the symmetric matrix $(B^{-1})'$ can be computed recursively by the procedure:
For $i = n-m, n-m-1, \ldots, 1$

$$\hat{b}'_{ii} = -\left(u_{ii}^{-2} u'_{ii} + \hat{b}_{ii} u'_{ii} + \sum_{k=i+1}^{\min\{i+m,n-m\}} (\hat{b}_{ik} u'_{ik} + \hat{b}'_{ik} u_{ik}) \right) / u_{ii}$$

For $j = i-1, i-2, \ldots, \max\{1, i-m\}$

$$\hat{b}'_{ij} = -\left(u'_{ii} \hat{b}_{ij} + \sum_{k=i+1}^{\min\{i+m,n-m\}} (u'_{ik} \hat{b}_{kj} + u_{ik} \hat{b}'_{kj}) \right) / u_{ii}$$

$$\hat{b}'_{ji} = \hat{b}'_{ij}$$

Since U and U' have bandwidth $m+1$, this procedure requires approximately $(2(m+1)^2 + 1)n$ operations (where the extra 1 is for $u_{ii}^{-2} u'_{ii}$, since u_{ii}^{-2}

is known from S^2 above). Thus, from (10), the central $2m+1$ band of $B^{-1}HB^{-1}$ can be computed efficiently. Lastly, since H is $(2m-1)$-banded, we can compute the diagonal of $B^{-1}HB^{-1}H$, and hence $\mathrm{tr}A^2(\lambda)$ from (9), in approximately $(m-1)n$ operations (utilizing the symmetry of $B^{-1}HB^{-1}$ and H).

Combining all the steps above, we have an $O(m^2n)$ algorithm for $\mathrm{tr}(I - A(\lambda))$ and $\mathrm{tr}A^2(\lambda)$, which requires approximately $(4(m+1)^2 + 2m + 3)n$ operations without the leverage values and $(5(m+1)^2+2m+3)n$ operations with the leverage values. When $m = 2$, these numbers are $43n$ and $52n$ operations, respectively.

4. Algorithms based on the log determinant

The algorithms in this section are based on differentiation of the log determinant of a matrix. Define $P = P(\lambda) = H + n\lambda G^TG$, which satisfies $P = \lambda B$. Then, from (3) and (4), it follows that $I - A(\lambda) = n\lambda GP^{-1}(\lambda)G^T$, and hence

$$\begin{aligned}\lambda^{-1}\mathrm{tr}(I - A(\lambda)) &= n\mathrm{tr}(GP^{-1}G^T) = n\mathrm{tr}(P^{-1}G^TG)\\ &= \mathrm{tr}(P^{-1}P') = \frac{d}{d\lambda}\log(\det P(\lambda)),\end{aligned} \tag{13}$$

where $P'(\lambda) = dP/d\lambda$ and the last equality is a general identity [28, p. 305]. Similarly, since $P''(\lambda) = 0$, we also have

$$\begin{aligned}\lambda^{-2}\mathrm{tr}(I - A(\lambda))^2 &= n^2\mathrm{tr}(P^{-1}G^TGP^{-1}G^TG)\\ &= -[\mathrm{tr}(P^{-1}P'') - \mathrm{tr}(P^{-1}P')^2] = -\frac{d^2}{d\lambda^2}\log(\det P(\lambda)),\end{aligned} \tag{14}$$

where the last equality is another general identity [28, p. 309].

The right hand sides of equations (13) and (14) can be computed efficiently as follows. We start by computing the Cholesky decomposition $P(\lambda) = LL^T$, where $L = L(\lambda)$ is lower triangular. This notation for the decomposition is used in this section to clearly distinguish it from the decomposition $B(p) = U^TU$ in Section 3. (Note that L and U are related by $L = \sqrt{\lambda}U^T$.) Differentiating $P(\lambda) = LL^T$ gives

$$P'(\lambda) = L'L^T + LL'^T = R_1, \tag{15}$$

where $L' = dL/d\lambda$ and $R_1 = nG^TG$. This provides a system of linear equations for the elements of L', which can be solved using the procedure in (11) and (12) in approximately $(m+1)^2n$ operations.

The second derivative of $P(\lambda)$ satisfies

$$P''(\lambda) = L''L^T + 2L'L'^T + LL''^T = 0,$$

so $(L''/2)L^T + L(L''/2)^T = R_2$, where $R_2 = -L'L'^T$, which is of the same form as (15). After computing the $(2m+1)$-banded symmetric matrix R_2 in approximately $(m+1)(m+2)n/2$ operations (by calculating the lower triangular part of $L'L'^T$ and using symmetry), we compute $L''/2$ in approximately $(m+1)^2 n$ operations by using the same procedure as for L' above. Then, the right hand sides of (13) and (14) are computed as

$$\frac{d}{d\lambda}\log(\det P(\lambda)) = \frac{d}{d\lambda}\log((\det L)^2) = 2\sum_{i=1}^{n-m}\frac{l'_{ii}}{l_{ii}},$$

$$-\frac{d^2}{d\lambda^2}\log(\det P(\lambda)) = -\frac{d^2}{d\lambda^2}\log((\det L)^2) = 2\sum_{i=1}^{n-m}\left(\frac{l'_{ii}}{l_{ii}}\right)^2 - 4\sum_{i=1}^{n-m}\frac{(l''_{ii}/2)}{l_{ii}}$$

(requiring about $3n$ operations), and, finally, we calculate

$$\mathrm{tr}A^2(\lambda) = n - 2\mathrm{tr}(I - A(\lambda)) + \mathrm{tr}(I - A(\lambda))^2.$$

Combining the steps above, we have an $O(m^2 n)$ algorithm, which requires approximately $((m+1)^2+1)n$ operations for $\mathrm{tr}(I - A(\lambda))$ only, and approximately $((5m+6)(m+1)/2+3)n$ operations for both $\mathrm{tr}(I - A(\lambda))$ and $\mathrm{tr}A^2(\lambda)$. When $m = 2$, these numbers are $10n$ and $27n$ operations, respectively.

5. Conclusions

First we compare the approaches in Sections 3 and 4 without computing leverage values. For the calculation of $\mathrm{tr}(I - A(\lambda))$, the algorithm in Section 4 (requiring approximately $((m+1)^2+1)n$ operations) is very close in efficiency to the algorithm in [12] (requiring approximately $((m+1)^2 n$ operations). For the calculation of both $\mathrm{tr}(I - A(\lambda))$ and $\mathrm{tr}A^2(\lambda)$, the algorithm in Section 4 (requiring approximately $((5m+6)(m+1)/2+3)n$ operations, i.e. $27n$ when $m = 2$) is significantly more efficient than the one in Section 3 (requiring approximately $(4(m+1)^2+2m+3)n$ operations, i.e. $43n$ when $m = 2$).

If leverage values are wanted in addition to both $\mathrm{tr}(I - A(\lambda))$ and $\mathrm{tr}A^2(\lambda)$, then one can use the corresponding complete algorithm in Section 3 (requiring approximately $(5(m+1)^2+2m+3)n$ operations, i.e. $52n$

when $m = 2$). Alternatively, one can combine the first part of it to generate the leverage values together with the complete algorithm in Section 4 (requiring approximately $(2(m+1)^2+(5m+6)(m+1)/2+m+6)n$ operations, i.e. $50n$ when $m = 2$).

For the calculation of both $\mathrm{tr}(I - A(\lambda))$ and $\mathrm{tr}A^2(\lambda)$ without leverage values, the algorithms in Sections 3 and 4 were implemented in MATLAB and tested on several examples with cubic splines ($m = 2$) and n up to 1600. It was found that they both performed very well in terms of speed and accuracy. For large n, the algorithm in Section 4 was noticeably faster than the one in Section 3, which is consistent with the operation counts above.

References

[1] R. L. Eubank, Spline Smoothing and Nonparametric Regression, Dekker, New York, 1988.

[2] G. Wahba, Spline Models for Observational Data, SIAM, Philadelphia, 1990.

[3] P. Craven, G. Wahba, Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross-validation, Numer. Math. 31 (1979) 377–403.

[4] T. Robinson, R. Moyeed, Making robust the cross-validatory choice of smoothing parameter in spline smoothing regression, Comm. Statist. Theory Methods 18 (1989) 523–539.

[5] M. A. Lukas, Robust generalized cross-validation for choosing the regularization parameter, Inverse Problems 22 (2006) 1883–1902.

[6] M. A. Lukas, Strong robust generalized cross-validation for choosing the regularization parameter, Inverse Problems 24 (2008) 034006.

[7] D. J. Cummins, T. G. Filloon, D. Nychka, Confidence intervals for nonparametric curve estimates: Toward more uniform pointwise coverage, J. Amer. Statist. Assoc. 96 (2001) 233–246.

[8] Y.-J. Kim, C. Gu, Smoothing spline Gaussian regression: more scalable computation via efficient approximation, J. Roy. Statist. Soc. Ser. B 66 (2004) 337–356.

[9] M. A. Lukas, F. R. de Hoog, R. S. Anderssen, Spline smoothing using robust GCV, Tech. Rep. 08-154, CMIS, CSIRO (2008).

[10] C. H. Reinsch, Smoothing by spline functions, Numer. Math. 10 (1967) 177–183.

[11] C. H. Reinsch, Smoothing by spline functions II, Numer. Math. 16 (1971) 451–454.

[12] M. F. Hutchinson, F. R. de Hoog, Smoothing noisy data with spline functions, Numer. Math. 47 (1985) 99–106.

[13] G. Wahba, Bayesian "confidence intervals" for the cross-validated smoothing spline, J. Roy. Statist. Soc. Ser. B 45 (1983) 133–150.

[14] T. Lyche, L. L. Schumaker, Computation of smoothing and interpolating natural splines via local bases, SIAM J. Numer. Anal. 10 (1973) 1027–1038.

[15] F. O'Sullivan, Discussion of "Some aspects of the spline smoothing approach to

non-parametric regression curve fitting" by B. W. Silverman, J. Roy. Statist. Soc. Ser. B 47 (1985) 39–40.
[16] C. Gu, Smoothing Spline ANOVA Models, Springer, New York, 2002.
[17] F. R. de Hoog, M. F. Hutchinson, An efficient method for calculating smoothing splines using orthogonal transformations, Numer. Math. 50 (1987) 311–319.
[18] L. Eldén, An algorithm for the regularization of ill-conditioned, banded least squares problems, SIAM J. Sci. Stat. Comput. 5 (1984) 237–254.
[19] F. R. de Hoog, R. S. Anderssen, M. A. Lukas, Differentiation of matrix functionals using triangular factorization, Tech. Rep. 09-46, CMIS, CSIRO (2009).
[20] P. Speckman, Spline smoothing and optimal rates of convergence in nonparametric regression models, Ann. Statist. 13 (1985) 970–983.
[21] B. W. Silverman, A fast and efficient cross-validation method for smoothing parameter choice in spline regression, J. Amer. Statist. Assoc. 79 (1984) 584–589.
[22] D. A. Girard, A fast "Monte-Carlo cross-validation" procedure for large least squares problems with noisy data, Numer. Math. 56 (1989) 1–23.
[23] M. F. Hutchinson, A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines, Comm. Statist. Simulation Comput. 18 (1989) 1059–1076.
[24] C. K. Carter, G. K. Eagleson, B. W. Silverman, A comparison of the Reinsch and Speckman splines, Biometrika 79 (1992) 81–91.
[25] A. Buja, T. Hastie, R. Tibshirani, Linear smoothers and additive models, Ann. Statist. 17 (2) (1989) 453–555.
[26] P. M. Anselone, P. J. Laurent, A general method for the construction of interpolating or smoothing spline-functions, Numer. Math. 12 (1968) 66–82.
[27] G. H. Golub, C. F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.
[28] D. A. Harville, Matrix Algebra from a Statistician's Perspective, New York: Springer, 1997.