View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Bath Research Portal

Citation for published version:
Wood, SN 2011, 'Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models', *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, vol. 73, no. 1, pp. 3-36. https://doi.org/10.1111/j.1467-9868.2010.00749.x

DOI:
10.1111/j.1467-9868.2010.00749.x

Publication date:
2011

Document Version
Peer reviewed version

Link to publication

The definitive version is available at www.blackwellsynergy.com

University of Bath

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 23. Jan. 2020

Fast stable REML and ML estimation of semiparametric GLMs

Simon N. Wood
Mathematical Sciences, University of Bath, Bath BA2 7AY U.K.
s.wood@bath.ac.uk

May 6, 2010

Abstract

Recent work by Reiss and Ogden (2009) provides a theoretical basis for sometimes preferring restricted maximum likelihood (REML) to generalized cross validation (GCV) for smoothing parameter selection in semiparametric regression. However, existing REML or marginal likelihood (ML) based methods for semiparametric GLMs use iterative REML/ML estimation of the smoothing parameters of working linear approximations to the GLM. Such indirect schemes need not converge, and fail to do so in a non-negligible proportion of practical analyses. By contrast, very reliable prediction error criteria smoothing parameter selection methods are available, based on direct optimization of GCV, or related criteria, for the GLM itself. Since such methods directly optimize properly defined functions of the smoothing parameters, they have much more reliable convergence properties. This article develops the first such method for REML or ML estimation of smoothing parameters. A Laplace approximation is used to obtain an approximate REML or ML for any GLM, which is suitable for efficient direct optimization. This REML/ML criterion requires that Newton-Raphson, rather then Fisher scoring, be used for GLM fitting, and a computationally stable approach to this is proposed. The REML or ML criterion itself is optimized by a Newton method, with the required derivatives obtained by a mixture of implicit differentiation and direct methods. The method will cope with numerical rank deficiency in the fitted model, and in fact provides a slight improvement in numerical robustness on the method of Wood (2008) for prediction error criteria based smoothness selection. Simulation results suggest that the new REML and ML methods offers some improvement in mean square error performance relative to GCV/AIC in most cases, without the small number of severe undersmoothing failures to which AIC and GCV are prone. This is achieved at the same computational cost as GCV/AIC. The new approach also eliminates the convergence failures of previous REML/ML based approaches for penalized GLMs, and usually has lower computational cost than these alternatives. Example applications are presented in adaptive smoothing, scalar on function regression and generalized additive model (GAM) selection.

Keywords: REML, Marginal Likelihood, GAMM, GAM, GCV, penalized GLM, penalized regression splines, stable computation, adaptive smoothing, scalar on function regression, model selection.

1 Introduction

This paper is about reliable and efficient computation of likelihood based smoothing parameter estimates in penalized generalized linear models (GLM). Consider a GLM in which n independent univariate response variables, y_i, with mean μ_i, depend on predictors via the model

$$g(\mu_i) = \mathbf{X}_i^* \boldsymbol{\beta}^* + \sum_j L_{ij} f_j, \quad y_i \sim \text{an exponential family distribution}, \tag{1}$$

where g is a known monotonic link function, the f_j are smooth but unknown functions of any number of covariates, the L_{ij} are known linear functionals (usually dependent on covariates), and $\mathbf{X}_i^*$ is the ith row of the model matrix for any strictly parametric model components, with corresponding coefficients $\boldsymbol{\beta}^*$. Restriction to the exponential family implies that $\text{var}(y_i) = \phi V(\mu_i)$, for some known 'variance function', V, and known or unknown 'scale parameter', ϕ. Typical $L_{ij} f_j$ terms are $f_j(x_i)$, $f_j(x_i) z_i$ or $\int f_j(x) k_i(x) dx$ (where k_i is known), corresponding to generalized additive, varying coefficient and signal regression models, respectively. For more on such models see, for example, Hastie and Tibshirani (1986, 1990); Ruppert, Wand and Carroll (2003); Wood (2006); Hastie and Tibshirani, (1993);

Marx and Eilers (1999); Ramsay and Silverman (2005); Reiss and Ogden (2007); Wahba (1990); Eilers and Marx (2002); Fahrmeir, Kneib and Lang (2004).

To estimate (1) in practice, the f_j can be represented by intermediate rank spline type basis expansions (as originally proposed by Wahba, 1980, and Parker and Rice, 1985, for example), in which case the model becomes the GLM (Nelder and Wedderburn, 1972)

$$g(\mu_i) = \mathbf{X}_i\boldsymbol{\beta}, \quad y_i \sim \text{an exponential family distribution}, \tag{2}$$

where $\boldsymbol{\beta}$ now includes $\boldsymbol{\beta}^*$ and all the basis coefficients, and $\mathbf{X}$ is the corresponding $n \times q$ model matrix, with q usually substantially less than n. If the spline bases dimensions are large enough to ensure reasonably low bias, then maximum likelihood estimation of (2) will almost certainly lead to overfitting. To avoid this, the model should be estimated by penalized likelihood maximization, where the penalties suppress overly wiggly components, f_j. In particular, the model is estimated by minimizing

$$D(\boldsymbol{\beta}) + \sum_j \lambda_j \boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}_j\boldsymbol{\beta} \tag{3}$$

w.r.t. $\boldsymbol{\beta}$, where D is the model deviance, defined as the saturated log likelihood minus the log likelihood, all multiplied by 2ϕ (D is a useful GLM analogue of the residual sum of squares of a linear model, and working in terms of D will allow the direct use of some results from Wood, 2008); the $\mathbf{S}_j$ are $q \times q$ positive semi-definite matrices and the λ_j are positive smoothing parameters. Usually the $\boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}_j\boldsymbol{\beta}$ measure the wiggliness of the f_j. In fact there may be several such penalties per f_j, for example when using tensor product (e.g. Wood, 2006) or adaptive (e.g. Krivobokova et al. 2008) smoothing bases. The $\mathbf{S}_j$ may also be components of more general random effects precision matrices.

Given the λ_j, there is a unique minimizer of (3), $\hat{\boldsymbol{\beta}}_\lambda$, which is straightforward to compute by a penalized version of the iteratively re-weighted least squares method used for GLM estimation (PIRLS, see e.g. Wood, 2006, or section 3.2). To select values for the λ_j requires optimization of a separate criteria, $\mathcal{V}(\boldsymbol{\lambda})$, say, which must be chosen.

1.1 Smoothness selection: prediction error or likelihood?

The λ_i selection criteria that have been proposed fall in to two main classes. The first group try to minimize model prediction error, by optimizing criteria such as Akaike's information criterion (AIC), cross validation or generalized cross validation (GCV) (see e.g. Wahba and Wold, 1975, Craven and Wahba, 1979). The second group treat the smooth functions as random effects (Kimeldorf and Wahba, 1970), so that the λ_i are variance parameters which can be estimated by maximum (marginal) likelihood (ML, Anderssen and Bloomfield, 1974) or restricted maximum likelihood/generalized maximum likelihood (REML/GML, Wahba, 1985).

Asymptotically prediction error methods give better prediction error performance than likelihood based methods (e.g. Wahba, 1985; Kauermann, 2005), but also have slower convergence of smoothing parameters to their optimal values (Härdle, Hall and Marron, 1988). Reflecting this, published simulation studies (e.g. Wahba, 1985; Gu, 2002; Ruppert, Wand and Carroll, 2003; Kohn, Ansley and Tharm, 1991), differ as to the relative performance of the two classes, although there is agreement that prediction error criteria are prone to occasional severe undersmoothing. Reiss and Ogden (2009) provide theoretical comparison of REML and GCV at*finite* sample sizes, showing that GCV is both more likely to develop multiple minima, and gives more variable λ_j estimates. Figure 1 illustrates the basic issue. GCV penalizes overfit only weakly, with a minimum that tends to be very shallow on the undersmoothing side, relative to sampling variability. This can lead to overfit. By contrast, REML (and also ML) penalizes overfit more severely, and therefor tends to have a much more pronounced optimum, relative to sampling variability. In principal, extreme undersmoothing can also be avoided by use of modified prediction error criteria such as AICc (Hurvich, Simonoff and Tsai, 1998), but in practice the use of low to intermediate rank bases for the f_j already suppresses severe overfit, and AICc then offers little *additional* benefit relative to GCV, as figure 1 also illustrates.

Greater resistance to overfit, less smoothing parameter variability and a reduced tendency to multiple minimia suggest that REML or ML might be preferable to GCV for semiparametric GLM estimation. But these benefits must be weighed against the fact that existing computational methods for REML/ML estimation of semiparametric GLMs are substantially less reliable than their prediction error equivalents, as the remainder of this section explains.

There are two main classes of computational method for λ_j estimation: those based on single iterations and those based on nested iterations. In the single iteration case, each PIRLS step, used to update $\hat{\boldsymbol{\beta}}$, is supplemented by a $\hat{\boldsymbol{\lambda}}$ update. The latter is based on improving a $\boldsymbol{\lambda}$ selection criteria $\mathcal{V}_{\hat{\beta}}(\lambda)$, which depends on the estimate of $\hat{\boldsymbol{\beta}}$ at the start of the step. $\mathcal{V}_{\hat{\beta}}(\boldsymbol{\lambda})$ will be some sort of REML, GCV or similar criterion, but it is not a fixed function of $\boldsymbol{\lambda}$, instead

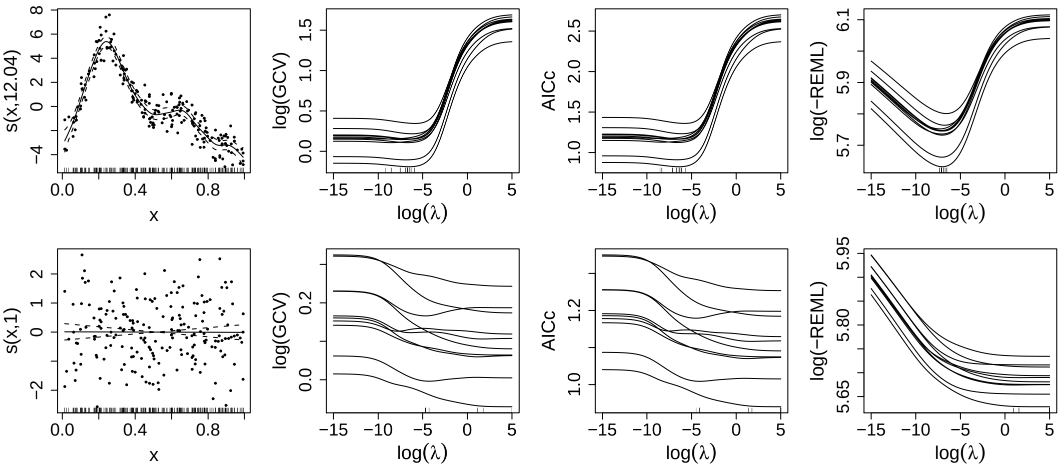

Figure 1: Example comparison of GCV, AICc and REML criteria (see section 1.1). The top left shows some x, y data modelled as $y_i = f(x_i) + \epsilon_i$, ϵ_i i.i.d. $N(0, \sigma^2)$ where smooth function f was represented using a rank 20 thin plate regression spline (Wood, 2003). The other panels on the top row plot various smoothness selection criteria against log smoothing parameters, for ten replicates of the data (each generated from the same 'truth'). Notice how shallow the GCV and AICc minima are relative to the sampling variability, resulting in rather variable optimal λ values (shown as a rug plot), and a propensity to undersmooth. In contrast the REML optima are much better defined, relative to the sampling variability, resulting in a smaller range of λ estimates. The bottom row is equivalent to the top row, but for data with no signal, so that the appropriate smoothing parameter should tend to infinity. Notice GCV's and AICc's occasional multiple minima and undersmoothing in this case, compared to the excellent behaviour of REML. Note that while AICc and GCV are not identical (compare the rug plots) AICc provides only marginal improvement on GCV. ML (not shown) has a similar shape to REML.

changing with $\hat{\boldsymbol{\beta}}$ from iterate to iterate. Consequently single iteration methods do not guarantee convergence to a fixed $\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\beta}}_{\hat{\lambda}}$ (see Gu, 2002, p.154; Wood, 2006, p. 180; Brezger, Kneib and Lang, 2007, Reference manual section 8.1.2).

In nested iteration, the smoothness selection criterion, $\mathcal{V}(\boldsymbol{\lambda})$, depends on $\boldsymbol{\beta}$ only via $\hat{\boldsymbol{\beta}}_{\lambda}$. An outer iteration updates $\hat{\boldsymbol{\lambda}}$ to optimize $\mathcal{V}(\boldsymbol{\lambda})$, with each iterative step requiring an inner PIRLS iteration to find the current $\hat{\boldsymbol{\beta}}_{\lambda}$. Because nested iteration optimizes a properly defined function of $\boldsymbol{\lambda}$, it is possible to guarantee convergence to a fixed optimum, provided that $\mathcal{V}$ is bounded below, and (3) has a well defined optimum (conditions which are rather mild, in practice). The disadvantage of nested iteration is substantially increased computational complexity.

To date only single iteration methods have been proposed for REML/ML estimation of semiparametric GLMs (e.g. Wood, 2004, using Breslow and Clayton, 1993, or Fahrmeir, Kneib and Lang, 2004, using Harville, 1977), and in practice convergence problems are not unusual: examples are provided in Wood (2004, 2008), and in Appendix A. Early prediction error based methods were also based on single iteration (e.g. Gu, 1992; Wood, 2004), and suffered similar convergence problems, but these were overcome by Wood's (2008) nested iteration method for GCV, GACV (generalized approximate cross validation) and AIC smoothness selection. Wood (2008) can not be extended to REML/ML while maintaining good numerical stability, so the purpose of this paper is to provide an efficient and stable nested iteration method for REML/ML smoothness selection, thereby removing the major practical obstacle to use of these criteria.

2 Approximate REML/ML for GLM smoothing parameter estimation

Since the work of Kimeldorf and Wahba (1970), Wahba (1983) and Silverman (1985), it has been recognized that the penalized likelihood estimates, $\hat{\boldsymbol{\beta}}$, are also the posterior modes of the distribution of $\boldsymbol{\beta}|\mathbf{y}$, if $\boldsymbol{\beta} \sim N(\mathbf{0}, \mathbf{S}^{-}\phi)$, where $\mathbf{S} = \sum_i \lambda_i \mathbf{S}_i$, and $\mathbf{S}^{-}$ is an appropriate generalized inverse (see e.g. Wood, 2006). Once the elements of $\boldsymbol{\beta}$ are viewed as random effects in this way, it is natural to try to estimate the λ_i, and possibly ϕ, by ML or REML (Wahba, 1985).

This preliminary section uses standard methods to obtain an approximate REML expression suitable for efficient direct optimization to estimate the smoothing parameters of a semi-parametric GLM. Rather than follow Patterson and

Thompson (1971) directly, Laird and Ware's (1982) approach to REML is taken, in which fixed effects are viewed as random effects with improper uniform priors, and integrated out. The key feature of the resulting expression is that it is relatively efficient to compute with, and is suitable for optimizing as a properly defined function of the smoothing parameters. That is, in contrast to previous single iteration approaches to this problem, there is no need to resort to optimizing the REML score of a working model. Since a very similar approach obtains an approximate ML, this is also derived. ML can be useful for comparing models with different smooth terms included, for example (REML can not be used for such comparison because the alternative models will differ in fixed effect structure).

Consider a penalized GLM with log likelihood $l(\boldsymbol{\beta}) = \log f_y(y|\boldsymbol{\beta})$. Under the random effects formulation we have an improper 'prior' density for $\boldsymbol{\beta}$,

$$f_\beta(\boldsymbol{\beta}) = \frac{|\mathbf{S}/\phi|_+^{0.5}}{\sqrt{2\pi}^{n_b - M_p}} \exp\{-\boldsymbol{\beta}^\mathsf{T}\mathbf{S}\boldsymbol{\beta}/(2\phi)\},$$

where $|\mathbf{B}|_+$ denotes the product of the non-zero eigenvalues of $\mathbf{B}$. n_b is the dimension of $\boldsymbol{\beta}$ and M_p is the dimension of the null space of $\mathbf{S}$. To obtain the restricted likelihood for REML we need to integrate $\boldsymbol{\beta}$ out of $f(y, \boldsymbol{\beta}) = f_y(y|\boldsymbol{\beta})f_\beta(\boldsymbol{\beta})$ (for ML we would need to integrate out the part of $\boldsymbol{\beta}$ that is in the range space of $\mathbf{S}$). In practice the integral can be approximated as follows. Let $\mathbf{H} = -\partial^2 l/\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^\mathsf{T}$, and $\hat{\boldsymbol{\beta}}$ be the maximizer of $f(y, \boldsymbol{\beta})$, that is the penalized likelihood estimates. Then

$$\begin{aligned} f(y,\boldsymbol{\beta}) &\simeq \exp\left[\log\{f_y(y|\hat{\boldsymbol{\beta}})\} + \log\{f_\beta(\hat{\boldsymbol{\beta}})\} - (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^\mathsf{T}(\mathbf{H} + \mathbf{S}/\phi)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})/2\right\} \\ &= f_y(y|\hat{\boldsymbol{\beta}})f_\beta(\hat{\boldsymbol{\beta}})\exp\{-(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^\mathsf{T}(\mathbf{H} + \mathbf{S}/\phi)(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})/2\}. \end{aligned}$$

Integrating w.r.t. $\boldsymbol{\beta}$, and denoting the likelihood by L, we get the Laplace approximate REML criterion

$$L_R(\lambda, \phi) = L(\hat{\boldsymbol{\beta}})f_\beta(\hat{\boldsymbol{\beta}})\frac{\sqrt{2\pi}^{n_b}}{|\mathbf{H} + \mathbf{S}/\phi|^{0.5}}$$

(actually exact for Gaussian models with the identity link). i.e. defining $l_r = \log L_r$,

$$2l_r = 2l(\hat{\boldsymbol{\beta}}) + \log|\mathbf{S}/\phi|_+ - \hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}\hat{\boldsymbol{\beta}}/\phi - \log|\mathbf{H} + \mathbf{S}/\phi| + M_p\log(2\pi).$$

If the penalized GLM has its coefficients estimated by Newton based PIRLS, as suggested below, then $\mathbf{H} = \mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}/\phi$, where $\mathbf{W}$ is a diagonal weight matrix. To get ML, rather than REML, we would need to re-parameterize to separate the parameters into penalized and unpenalized. Then $\mathbf{H}$ would be the negative Hessian for the penalized parameters only: further details are provided below in section 2.1.

For ease of computation it helps to separate out l_r into ϕ dependent and ϕ independent components. To this end, let $l_s(\phi)$ denote the saturated log likelihood and define

$$D_p = D(\hat{\boldsymbol{\beta}}) + \hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}\hat{\boldsymbol{\beta}}$$

and (assuming Newton weights)

$$K = (\log|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S}| - \log|\mathbf{S}|_+)/2.$$

We then have that

$$-l_r = \frac{D_p}{2\phi} - l_s(\phi) + K - \frac{M_p}{2}\log(2\pi\phi). \tag{4}$$

There are two approaches to the estimation of ϕ: (i) estimate ϕ as part of l_r maximization, or (ii) use the Pearson statistic over $n - M_p$ as $\hat{\phi}$, and optimize the resulting criterion, taking account of the derivatives of $\hat{\phi}$ w.r.t. the smoothing parameters. The only advantage of (ii) is that it may sometimes allow the resulting REML score to be used as a heuristic method of smoothness selection with quasi-likelihood.

The simpler approach of using the expected Hessian in place of $\mathbf{H}$ was also investigated, but in simulations gave worse performance than GCV when non-canonical links were used.

2.1 ML details

For Laplace approximate ML, rather than REML, estimation, the only difference to the criterion is that we now need $\mathbf{H}$ to be the negative Hessian w.r.t. the coefficients of any orthogonal basis for the range space of the penalty. The easiest way to separate out the range space is to form the eigen-decomposition $\sum_j \mathbf{S}_j / \|\mathbf{S}_j\|_F = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\mathsf{T}$, where the scaling of each $\mathbf{S}_j$ by its Frobenious norm maintains good numerical conditioning. The first $q - M_p$ columns of $\mathbf{U}$ now form an orthogonal basis for the range space of $\mathbf{S}$ (see e.g. Wood, 2006, section 4.8.2 and 6.6.1). In consequence, if we re-parameterize by setting $\bar{\boldsymbol{\beta}} = \mathbf{U}^\mathsf{T}\boldsymbol{\beta}$ then the first $q - M_p$ elements of $\bar{\boldsymbol{\beta}}$ will be penalized and should be integrated out of the joint density of $\mathbf{y}$ and $\bar{\boldsymbol{\beta}}$, while the last M_p elements are unpenalized, and hence left alone. Let $\mathbf{U}_1$ be the first $q - M_p$ columns of $\mathbf{U}$. Applying the re-parameterization we have $\bar{\mathbf{X}} = \mathbf{X}\mathbf{U}_1$ and $\bar{\mathbf{S}} = \mathbf{U}_1^\mathsf{T}\mathbf{S}\mathbf{U}_1$, and some work establishes that the negative (Laplace approximate) log marginal likelihood is

$$-l = \frac{D_p}{2\phi} - l_s(\phi) + (\log|\bar{\mathbf{X}}^\mathsf{T}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}| - \log|\mathbf{S}|_+)/2. \qquad (5)$$

2.2 Accuracy of the Laplace approximation

For fixed dimension of $\boldsymbol{\beta}$, the true REML or ML integral divided by its Laplace approximation is $1 + O(n^{-1})$ (see e.g. Davison, 2003, section 11.3.1). For consistency, it is usually necessary for the dimension of $\boldsymbol{\beta}$ to grow with n, which reduces this rate somewhat. However, for spline type smoothers the dimension need only grow slowly with n (e.g. Gu and Kim, 2002, show that the rate need only be $O(n^{2/9})$ for cubic spline like smooths), so that convergence is still rapid. Kauermann et al. (2009) show in detail that the Laplace approximation is well justified asymptotically for ML in the penalized regression spline setting.

Rapid convergence does not in itself guarantee that the approximation is sufficiently accurate for any particular finite sample. Fortunately a simple and computationally efficient accuracy check is readily implemented, since a rather precise unbiased estimator of the REML score can be obtained by importance sampling with a 'Laplace proposal'. In particular, if $\mathbf{R}$ is a square factor such that $\mathbf{R}^\mathsf{T}\mathbf{R} = (\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S})^{-1}\hat{\phi}$, and $\mathbf{z}_i$ are n_s independent $N(\mathbf{0}, \mathbf{I})$ random n_b vectors, then

$$\frac{(2\pi)^{n_b/2}}{n_s|\mathbf{R}|}\sum_{i=1}^{n_s} f_y(y|\hat{\boldsymbol{\beta}} + \mathbf{R}^\mathsf{T}\mathbf{z}_i) f_\beta(\hat{\boldsymbol{\beta}} + \mathbf{R}^\mathsf{T}\mathbf{z}_i) e^{\|\mathbf{z}_i\|^2/2}$$

is an unbiased estimator of the exact REML score (see, for example, Monahan, 2001, section 10.4C). In the work reported here n_s in the range 1000 to 10000 was sufficient to ensure that the Monte-Carlo variability was at least an order of magnitude smaller than the mean difference between the estimator and the deterministic Laplace approximation. This estimator was used to estimate the Laplace approximation error, at the estimated smoothing parameters, for all the examples presented subsequently in this paper. The worst error was for the binary simulations in section 4, where the magnitude of the error was up to 0.3. The other examples had approximation errors an order of magnitude smaller. Hence the error induced by the deterministic Laplace approximation is not significant relative to the sampling uncertainty in the smoothing parameters, suggesting that the Laplace approximation is adequate for the examples presented here.

Note that the Laplace approximation employed here does not suffer from the difficulties common to most PQL (Breslow and Clayton, 1993) implementations when used with binary data. Most PQL implementations have to estimate ϕ for the working model, even with binary data where this is not really satisfactory. In addition, PQL uses the expected Hessian in place of the exact Hessian when non-canonical links are used, which also reduces accuracy. That said, it should still expected that the accuracy of (4) and (5) will reduce for binary or Poisson data when the expectation of the response variable is very low.

3 Optimizing the REML criterion

(4) and (5) depend on the smoothing parameter vector, $\boldsymbol{\lambda}$, via the dependence of $\mathbf{S}$, $\hat{\boldsymbol{\beta}}$ (and hence $\mathbf{W}$) on $\boldsymbol{\lambda}$. The proposal here is to optimize (4) or (5) w.r.t. the $\rho_i = \log(\lambda_i)$, using Newton's method, with the usual modifications that (i) some step length control will be used and (ii) the Hessian will be perturbed to be positive definite, if it is not (see Nocedal and Wright, 2006, for an up to date treatment and computational details). Each trial log smoothing parameter vector, $\boldsymbol{\rho}$, proposed as part of the Newton method iteration, will require a PIRLS iteration to evaluate the corresponding $\hat{\boldsymbol{\beta}}$ (and hence $\mathbf{W}$). So the whole optimization consists of two nested iterations: an outer to find $\hat{\boldsymbol{\rho}}$, and

an inner to find the $\hat{\boldsymbol\beta}$ corresponding to any $\boldsymbol\rho$. The outer iteration requires the gradient and Hessian of (4) or (5) w.r.t. $\boldsymbol\rho$, and this in turn requires first and second derivatives of $\hat{\boldsymbol\beta}$ w.r.t. $\boldsymbol\rho$.

Irrespective of the details of the optimization method, the major difficulty in minimizing (4) or (5) is that if some λ_j is large enough, then the 'numerical footprint' of the corresponding penalty term $\lambda_j \boldsymbol\beta^{\mathsf T}\mathbf{S}_j\boldsymbol\beta$ can extend well beyond the penalty's range space: i.e. numerically the penalty can have marked effects in the subspace of the model parameter space for which, formally, $\boldsymbol\beta^{\mathsf T}\mathbf{S}_j\boldsymbol\beta = 0$. For example if $\|\lambda_j\mathbf{S}_j\| \gg \|\lambda_k\mathbf{S}_k\|$ then $\lambda_j\mathbf{S}_j$ can have effects which are 'numerically zero' when judged relative to $\|\lambda_j\mathbf{S}_j\|$ (and would be exactly zero in infinite precision arithmetic), but which are larger than the strictly non-zero effects of $\lambda_k\mathbf{S}_k$. If left uncorrected, this problem leads to serious errors in evaluation of $\hat{\boldsymbol\beta}$, $|\mathbf{S}|_+$ and $|\mathbf{X}^{\mathsf T}\mathbf{W}\mathbf{X} + \mathbf{S}|$ and their derivatives w.r.t. $\boldsymbol\rho$ (see section 3.1). Because multiple penalties often have overlapping range spaces (i.e. they penalize intersecting subspaces of the parameter space), no single re-parameterization can solve this problem for all $\boldsymbol\lambda$ values, but an adaptive reparametrization approach does work, and is outlined in section 3.1. Note that the Wood (2008) method, for dealing with numerical ill-conditioning for prediction error criteria, is hopeless here. That method truncates the parameter space to deal with ill-conditioning induced by changes in $\boldsymbol\lambda$, but such an approach would lead to large erroneous and discontinuous changes in $|\mathbf{S}|_+$ and $|\mathbf{X}^{\mathsf T}\mathbf{W}\mathbf{X}+\mathbf{S}|$ as $\boldsymbol\lambda$ changes. We will of course still need to truncate the parameter space if some parameters would not be identifiable whatever the value of $\boldsymbol\lambda$, but such a $\boldsymbol\lambda$ independent truncation is not problematic.

A second question, when minimizing (4) or (5), is what optimization method to use to obtain the $\hat{\boldsymbol\beta}_\lambda$ corresponding to any trial $\boldsymbol\lambda$? If a PIRLS scheme is employed based on Newton (rather than Fisher) updates, then the Hessian required in (4) or (5) is conveniently obtained as a by-product of fitting, which also means that the same method can be used to stabilize both $\hat{\boldsymbol\beta}$ and REML/ML evaluation. Furthermore the required derivatives of $\hat{\boldsymbol\beta}$ w.r.t. $\boldsymbol\rho$ can be obtained directly from the information available as part of the PIRLS, using implicit differentiation, without the need for further iteration. Newton based PIRLS also leads to more rapid convergence with non-canonical links.

As a result of the preceding considerations, this paper proposes that the following steps should be taken for each trial $\boldsymbol\rho$ proposed by the outer Newton iteration.

1. Reparameterize to avoid large norm $\lambda_j\mathbf{S}_j$ terms having effects outside their range spaces, thereby ensuring accurate computation with the current $\boldsymbol\rho$. (Section 3.1.)

2. Estimate $\hat{\boldsymbol\beta}$ by Newton based PIRLS, setting to zero any elements of $\hat{\boldsymbol\beta}$ which would be unidentifiable *irrespective of the value of* $\boldsymbol\rho$. (Sections 3.2 and 3.3.)

3. Obtain first and second derivatives of $\hat{\boldsymbol\beta}$ w.r.t. $\boldsymbol\rho$, using implicit differentiation and the quantities calculated as part of step 2. (Section 3.4.)

4. Using the results from parts 2 and 3, evaluate the REML/ML criterion and derivatives w.r.t. $\boldsymbol\rho$. (Section 3.5.)

After these four steps, all the ingredients are in place to propose a new $\boldsymbol\rho$ using a further step of Newton's method.

3.1 Re-parameterization, $\log|\mathbf{S}|_+$ and $\sqrt{\mathbf{S}}$

$\log|\mathbf{S}|_+$ (where $\mathbf{S} = \sum_j \lambda_j\mathbf{S}_j$) is the most numerically troublesome term in the REML/ML objective. Both $\lambda_i \to 0$ *and* $\lambda_i \to \infty$ can cause numerical problems when evaluating the determinant. The problem is most easily seen by considering the simple example of evaluating $|\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2|$ when the $q \times q$ positive semi-definite matrices $\mathbf{S}_j$ are not full rank, but $\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2$ is. In what follows let $\|\cdot\|$ denote the matrix 2-norm (although the 1, ∞ or Frobenious norms would serve as well), and let $\hat{x}$ denote the computed version of any quantity x. Consider a similarity transform based on the eigen decomposition $\mathbf{S}_1 = \mathbf{U}\boldsymbol\Lambda\mathbf{U}^{\mathsf T}$, with computed version $\mathbf{S}_1 = \hat{\mathbf{U}}\hat{\boldsymbol\Lambda}\hat{\mathbf{U}}^{\mathsf T}$. Let $\boldsymbol\Lambda^+$ denote the vector of strictly positive eigenvalues, and $\boldsymbol\Lambda^0$ the vector of zero eigenvalues, and note that $\hat{\boldsymbol\Lambda}^0$ will have elements of typical magnitude $\|\mathbf{S}_1\|\epsilon_m$ where ϵ_m is the computational machine precision (see e.g. Watkins, 1991, Section 5.5 or Golub and van Loan, 1996, Chapter 8).

By standard properties of similarity transforms we have

$$|\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2| = |\lambda_1\boldsymbol\Lambda + \lambda_2\mathbf{U}^{\mathsf T}\mathbf{S}_2\mathbf{U}|. \tag{6}$$

Suppose that $\mathbf{S}_j$ has rank r_j and rank deficiency $d_j = q - r_j$. As $\lambda_1/\lambda_2 \to \infty$ it is routine that the r_1 largest eigenvalues of $\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2 \to \lambda_1\boldsymbol\Lambda^+$, so that $|\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2| \to \lambda_1^{r_1}\prod_i \Lambda_i^+\alpha$, where the factor α depends on $\lambda_2\mathbf{S}_2$. However as $\lambda_1/\lambda_2 \to \infty$ *all* the *computed* eigenvalues of $\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2 \to \lambda_1\hat{\boldsymbol\Lambda}$, so that $|\lambda_1\widehat{\mathbf{S}_1 + \lambda_2}\mathbf{S}_2| \to \lambda_1^{r_1}\prod_i \hat{\Lambda}_i^+\lambda_1^{d_1}\prod_i \hat{\Lambda}_i^0$.

Hence the computed determinant is seriously in error because the factor $\lambda_1^{d_1}\prod_i \hat{\Lambda}_i^0$ is essentially arbitrary, and is unrelated to the correct factor α. (Notice that the problem vanishes for a full rank $\mathbf{S}_1$.)

The difficulty arises because the computed version of the matrix $\lambda_1\mathbf{\Lambda} + \lambda_2\mathbf{U}^\mathsf{T}\mathbf{S}_2\mathbf{U}$ is perturbed by the completely arbitrary error terms in $\lambda_1\hat{\mathbf{\Lambda}}^0$. In general the effect of a perturbation on the determinant of a positive definite $\mathbf{A}$, with eigenvalues $\mathbf{\Lambda}^A$, depends on the size of the perturbation relative to $\min(\mathbf{\Lambda}^A)$. This is easily seen by considering a simple additive perturbation $\epsilon\mathbf{I}$ (where ϵ is the perturbation size). Then $|\mathbf{A} + \epsilon\mathbf{I}|/|\mathbf{A}| = \prod_i(\Lambda_i^A + \epsilon)/\Lambda_i^A$, where the largest contribution to the right hand side is from the term $\{\min(\mathbf{\Lambda}^A) + \epsilon\}/\min(\mathbf{\Lambda}^A)$. Hence we can expect problems when the perturbations, $\lambda_1\hat{\mathbf{\Lambda}}^0$, become non-negligible relative to the smallest eigenvalue of $\lambda_1\mathbf{S}_1 + \lambda_2\mathbf{S}_2$, which is bounded below by the smallest positive eigenvalue of $\lambda_2\mathbf{S}_2$ as $\lambda_1/\lambda_2 \to \infty$.

In short, we can expect this 'numerical zero leakage' issue to spoil determinant calculations whenever the ratio of the largest strictly positive eigenvalue of $\lambda_1\mathbf{S}_1$ (which sets the scale of the arbitrary perturbation, $\lambda_1\hat{\mathbf{\Lambda}}^0$) to the smallest strictly positive eigenvalue of $\lambda_2\mathbf{S}_2$ is too great. However, the example also suggests a simple way of suppressing the problem. Re-parameterize using the computed eigenbasis of the dominant term $\mathbf{S}_1$, so that $\mathbf{S}_1$ becomes $\hat{\mathbf{\Lambda}}$ and $\mathbf{S}_2$ becomes $\hat{\mathbf{U}}^\mathsf{T}\mathbf{S}_2\hat{\mathbf{U}}$. In the transformed space it is easy to ensure that the dominant term (now $\hat{\mathbf{\Lambda}}$) only acts within its range space, by setting $\hat{\mathbf{\Lambda}}^0 = \mathbf{0}$ (if the rank of $\mathbf{S}_1$ is known then identifying which eigenvalues should be zero is trivial, if not, see step 3 in Appendix B).

Having re-parameterized and truncated in this way, stable evaluation of $|\lambda_1\mathbf{\Lambda} + \lambda_2\mathbf{U}^\mathsf{T}\mathbf{S}_2\mathbf{U}|$ is straightforward. Only the first r_1 columns of $\lambda_1\hat{\mathbf{\Lambda}} + \lambda_2\hat{\mathbf{U}}^\mathsf{T}\mathbf{S}_2\hat{\mathbf{U}}$, now depend on $\lambda_1\mathbf{S}_1$. Forming a pivoted QR decomposition $\lambda_1\hat{\mathbf{\Lambda}} + \lambda_2\hat{\mathbf{U}}^\mathsf{T}\mathbf{S}_2\hat{\mathbf{U}} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ maintains this column separation in $\hat{\mathbf{R}}$ (the decomposition acts on columns, without mixing between columns), with the result that $|\lambda_1\widehat{\mathbf{S}_1 + \lambda_2\mathbf{S}_2}| = |\lambda_1\hat{\mathbf{\Lambda}} + \lambda_2\hat{\mathbf{U}}^\mathsf{T}\mathbf{S}_2\hat{\mathbf{U}}| = \prod_i \hat{R}_{ii}$ can be accurately computed. Furthermore, pivoting ensures that $\hat{\mathbf{R}}^{-1}$ is computable, which is necessary for derivative calculations. See Golub and van Loan (1996) for full discussion of QR decomposition with pivoting.

The stable computation of $\hat{\boldsymbol{\beta}}$, discussed in section 3.3, will also require that a square root of $\mathbf{S}$ can be formed that maintains the required 'column separation' of the dominant terms in $\mathbf{S}$ (that is, we must not end up with large magnitude elements in some column $j > r_1$, just because $\lambda_1\|\mathbf{S}_1\|$ is large). This is quite straightforward under the reparameterization just discussed. For example, let $\hat{\mathbf{S}}' = \lambda_1\hat{\mathbf{\Lambda}} + \lambda_2\hat{\mathbf{U}}^\mathsf{T}\mathbf{S}_2\hat{\mathbf{U}}$ (with $\hat{\mathbf{\Lambda}}$'s 'machine zeros' set to true zeros) and $\hat{\mathbf{P}}$ be the diagonal matrix such that $\hat{P}_{ii} = \sqrt{|\hat{S}'_{ii}|}$. Forming the Choleski decomposition $\hat{\mathbf{L}}\hat{\mathbf{L}}^\mathsf{T} = \hat{\mathbf{P}}^{-1}\hat{\mathbf{S}}'\hat{\mathbf{P}}^{-1}$, then $\hat{\mathbf{E}} = \hat{\mathbf{L}}^\mathsf{T}\hat{\mathbf{P}}$ is a matrix square root such that $\hat{\mathbf{E}}^\mathsf{T}\hat{\mathbf{E}} = \hat{\mathbf{S}}'$. Furthermore, $\lambda_1\mathbf{S}_1$ only affects the size of the elements in $\hat{\mathbf{E}}$'s first r_1 columns (this is easily seen, since, from the definition of $\hat{\mathbf{E}}$, the squared Euclidian norm of its j^{th} column is given by $\hat{\mathbf{S}}'_{jj}$, which does not depend on $\lambda_1\mathbf{S}_1$ if $j > r_1$). The preconditioning (or 'scaling') matrix, $\hat{\mathbf{P}}^{-1}$, ensures that the Choleski factor can be computed in finite precision, however divergent the sizes of the components of $\mathbf{S}$ (see e.g. Watkins, 1991, Section 2.9). From now on no further purpose is served by distinguishing between 'true' and computed quantities, so hats will be omitted.

Of course $\mathbf{S} = \sum \lambda_i\mathbf{S}_i$ generally contains more than two terms and is not full rank, but appendix B generalizes the similarity transform based reparameterization, along with the (generalized) determinant and square root calculations, to any number of components of a rank deficient $\mathbf{S}$. It also provides the expressions for the derivatives of $\log|\mathbf{S}|_+$ w.r.t. $\boldsymbol{\rho}$. The operations count for appendix B is $O(q^3)$.

The stable matrix square root, $\mathbf{E}$, produced by the Appendix B method, is only useful if the rest of the model fitting adopts the Appendix B re-parameterization. That is, the transformed $\mathbf{S}_i$, $\mathbf{S}$ and $\mathbf{E}$, computed by Appendix B, must be used in place of the original untransformed versions, along with a transformed version of the model matrix. To compute the latter, let $\mathbf{Q}_s$ be the orthogonal matrix describing the similarity transform applied by Appendix B. i.e. if $\mathbf{S}$ is the transformed total penalty matrix, then formally, $\mathbf{Q}_s\mathbf{S}\mathbf{Q}_s^\mathsf{T}$ is the untransformed original. Then the transformed model matrix should be $\mathbf{X}\mathbf{Q}_s$ (obtained at $O(nq^2)$ cost). In what follows it is assumed that this re-parameterization is always adopted, being re-computed for each new $\boldsymbol{\rho}$ value. So the model matrix and penalty matrices are taken to be the transformed versions, from now on. Note that if the coefficient estimates in this parameterization are $\hat{\boldsymbol{\beta}}$, then the estimates in the original parameterization are $\mathbf{Q}_s\hat{\boldsymbol{\beta}}$.

Finally, note that re-parameterization is preferable to simply limiting the working $\boldsymbol{\lambda}$ range. To keep the non zero eigenvalues of all $\lambda_i\mathbf{S}_i$ within limits that guarantee computational stability, usually entails unacceptably restrictive limits on the λ_i. i.e. limits restrictive enough to ensure numerical stability have statistically noticeable effects.

3.2 Estimating the regression coefficients given smoothing parameters

Minimizing (3) by Newton's method or Fisher scoring both result in a penalized iteratively re-weighted least squares method, as follows. Pseudodata and weights are defined first:

$$z_i = \eta_i + (y_i - \mu_i)g'_i/\alpha_i \quad \text{and} \quad w_i = \frac{\omega_i \alpha_i}{V_i g_i'^2}$$

where $\eta_i = g(\mu_i) = \mathbf{X}_i\boldsymbol{\beta}$, $V_i = V(\mu_i)$,

$$\alpha_i = \begin{cases} 1 + (y_i - \mu_i)\left(\frac{V'_i}{V_i} + \frac{g''_i}{g'_i}\right) & \text{for Newton's method} \\ 1 & \text{for Fisher scoring} \end{cases}$$

and x' denotes $\mathrm{d}x/\mathrm{d}\mu_i$, whatever x. These quantities are always evaluated at the current μ_i estimates. The ω_i are any prior weights, and are usually 1. If a canonical link function is used then $\alpha_i = 1 \; \forall \; i$ and Newton's method and Fisher scoring coincide.

Estimation of the coefficients, $\boldsymbol{\beta}$, is performed by the modified PIRLS scheme of iterating the following two steps to convergence ($\boldsymbol{\mu}$ estimates are initialized using the previous $\hat{\boldsymbol{\beta}}_\lambda$, or directly from $\mathbf{y}$).

1. Given the current *estimate* of $\boldsymbol{\mu}$ (and hence $\boldsymbol{\eta}$), evaluate $\mathbf{z}$ and $\mathbf{w}$.

2. Solve the weighted penalized least squares problem of minimizing

$$\sum_{i=1}^{n} w_i(z_i - \mathbf{X}_i\boldsymbol{\beta})^2 + \sum_j \lambda_j \boldsymbol{\beta}^{\mathsf{T}}\mathbf{S}_j\boldsymbol{\beta} \tag{7}$$

w.r.t. $\boldsymbol{\beta}$, to obtain the updated estimate of $\boldsymbol{\beta}$ and hence $\boldsymbol{\mu}$ (and $\boldsymbol{\eta}$). See section 3.3.

At convergence of the Newton type iteration the Hessian of the deviance w.r.t. $\boldsymbol{\beta}$ is given by $2\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}$, where $\mathbf{W} = \mathrm{diag}(w_i)$. Under Fisher scoring $2\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}$ is the *expected* Hessian. See e.g. Green and Silverman (1994) or Wood (2006) for further information on (Fisher based) penalized iteratively reweighted least squares.

Several points should be noted. (i) Step halving will be needed in the event that the penalized deviance increases at any iteration, but the Newton method should never require it at the end of the iteration. (ii) The Newton scheme tends to converge faster than Fisher scoring in non-canonical link situations, an effect which can be particularly marked when using Tweedie (1984) distributions. (iii) With non-canonical links, the w_i need not all be positive for the Newton scheme, and in practice negative weights are encountered for perfectly reasonable models: the next section deals with this. Negative w_i provide the second reason that the Wood (2008) method can not be extended to REML.

3.3 Stable least squares with negative weights

This section develops a method for stable computation of weighted least squares problems when some weights are negative, as required by the Newton based PIRLS described in section 3.2. The method also deals with identifiability problems that do not depend on the magnitude of $\boldsymbol{\lambda}$.

The obvious approach to solving (7) in the presence of negative weights would be to directly solve

$$(\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} + \mathbf{S})\hat{\boldsymbol{\beta}} = \mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{z} \tag{8}$$

for $\hat{\boldsymbol{\beta}}$, where $\mathbf{W} = \mathrm{diag}(w_i)$, $\mathbf{z}$ is the vector of z_i from section 3.2 and $\mathbf{S} = \sum_j \lambda_j \mathbf{S}_j$. However, it is well known that direct formation of $\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}$ results in a system with a condition number that is the square of what is necessary (see e.g. Golub and van Loan, 1996, sections 5.3.2 and 5.3.8). Given that penalized GLMs are frequently complex models in which concurvity effects can easily lead to quite high condition numbers, this approach is not sensible.

When weights are non negative, stable solution of (8) is based on orthogonal decomposition of $\sqrt{\mathbf{W}}\mathbf{X}$ (e.g. Wood, 2004), but this does not work if some weights are negative. This section proposes a stable solution method, by starting with a 'nearby' penalized least squares problem, for which all the weights are non-negative, applying a stable orthogonal decomposition approach to this, but at the same time developing the correction terms necessary to end up with the solution to (8) itself.

To make progress then, let $\mathbf{W}^-$ denote the diagonal matrix such that $W^-_{ii} = 0$ if $w_i \geq 0$ and $-w_i$ otherwise. Also let $\bar{\mathbf{W}}$ be a diagonal matrix with $\bar{W}_{ii} = |w_i|$. In this case

$$\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} = \mathbf{X}^{\mathsf{T}}\bar{\mathbf{W}}\mathbf{X} - 2\mathbf{X}^{\mathsf{T}}\mathbf{W}^-\mathbf{X}.$$

So $\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}$ has been split into a component that is straightforward to compute with stably, and a 'correction' term. Starting with the straightforward term, perform a QR decomposition

$$\sqrt{\bar{\mathbf{W}}}\mathbf{X} = \boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{R}} \tag{9}$$

(either without pivoting, or reversing the pivoting of $\boldsymbol{\mathcal{R}}$ after the decomposition). At this stage it is necessary to test for any inherent lack of identifiability in the problem (that is lack of identifiability which is $\boldsymbol{\lambda}$ independent). Section 3.3.1 describes how to do this. For the moment suppose that the inherent rank of the problem is r, and we have a list of any unidentifiable parameters. Then drop the columns of $\boldsymbol{\mathcal{R}}$ and $\mathbf{X}$ and the rows and columns of the $\mathbf{S}_i$ corresponding to any unidentifiable parameters.

$\boldsymbol{\mathcal{R}}$ is now a square root of $\mathbf{X}^{\mathsf{T}}\bar{\mathbf{W}}\mathbf{X}$, but we really need a square root of $\mathbf{X}^{\mathsf{T}}\bar{\mathbf{W}}\mathbf{X} + \mathbf{S}$, in order to move towards solution of (8). To this end, let $\mathbf{E}$ be a matrix such that $\mathbf{E}^{\mathsf{T}}\mathbf{E} = \mathbf{S}$, computed as described in Appendix B and section 3.1. Drop the columns of $\mathbf{E}$ corresponding to any unidentifiable parameters, and form a further pivoted QR decomposition

$$\begin{pmatrix} \boldsymbol{\mathcal{R}} \\ \mathbf{E} \end{pmatrix} = \mathbf{Q}\mathbf{R}. \tag{10}$$

$\mathbf{R}$ is the required square root of $\mathbf{X}^{\mathsf{T}}\bar{\mathbf{W}}\mathbf{X} + \mathbf{S}$. Now define $n \times r$ matrix $\mathbf{Q}_1 = \boldsymbol{\mathcal{Q}}\mathbf{Q}[1:q,]$, where q is the number of columns of $\mathbf{X}$ and $\mathbf{Q}[1:q,]$ denotes the first q rows of $\mathbf{Q}$. Hence

$$\sqrt{\bar{\mathbf{W}}}\mathbf{X} = \mathbf{Q}_1\mathbf{R}. \tag{11}$$

For what follows, the pivoting used in the QR step (10) will have to be applied to the rows and columns of $\mathbf{S}_j$ and the columns of $\mathbf{X}$.

Now we need to correct the matrix square root $\mathbf{R}$ to obtain what is actually needed to solve (8):

$$\begin{aligned}
\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} + \mathbf{S} &= \mathbf{R}^{\mathsf{T}}\mathbf{R} - 2\mathbf{X}^{\mathsf{T}}\mathbf{W}^-\mathbf{X} \\
&= \mathbf{R}^{\mathsf{T}}(\mathbf{I} - 2\mathbf{R}^{-\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{W}^-\mathbf{X}\mathbf{R}^{-1})\mathbf{R} \\
&= \mathbf{R}^{\mathsf{T}}(\mathbf{I} - 2\mathbf{R}^{-\mathsf{T}}\mathbf{R}^{\mathsf{T}}\mathbf{Q}_1^{\mathsf{T}}\mathbf{I}^-\mathbf{Q}_1\mathbf{R}\mathbf{R}^{-1})\mathbf{R} \\
&= \mathbf{R}^{\mathsf{T}}(\mathbf{I} - 2\mathbf{Q}_1^{\mathsf{T}}\mathbf{I}^-\mathbf{Q}_1)\mathbf{R},
\end{aligned}$$

where $\mathbf{I}^-$ denotes the diagonal matrix such that $I^-_{ii} = 0$ if $w_i > 0$ and 1 otherwise, while $\mathbf{W}^- = \mathbf{I}^-\bar{\mathbf{W}}$. The matrix $\mathbf{I} - 2\mathbf{Q}_1^{\mathsf{T}}\mathbf{I}^-\mathbf{Q}_1$ is not necessarily positive semi definite, and so requires careful handling. Forming the singular value decomposition

$$\mathbf{I}^-\mathbf{Q}_1 = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathsf{T}} \tag{12}$$

(of course, in practice the zero rows of $\mathbf{I}^-\mathbf{Q}_1$ can be dropped before decomposition) then we obtain

$$\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} + \mathbf{S} = \mathbf{R}^{\mathsf{T}}(\mathbf{I} - 2\mathbf{V}\mathbf{D}^2\mathbf{V}^{\mathsf{T}})\mathbf{R} = \mathbf{R}^{\mathsf{T}}\mathbf{V}(\mathbf{I} - 2\mathbf{D}^2)\mathbf{V}^{\mathsf{T}}\mathbf{R} \tag{13}$$

(and additionally $\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} = \boldsymbol{\mathcal{R}}^{\mathsf{T}}\boldsymbol{\mathcal{R}} - 2\mathbf{R}^{\mathsf{T}}\mathbf{V}\mathbf{D}^2\mathbf{V}^{\mathsf{T}}\mathbf{R}$). Now define

$$\mathbf{P} = \mathbf{R}^{-1}\mathbf{V}(\mathbf{I} - 2\mathbf{D}^2)^{-1/2} \quad \text{and} \quad \mathbf{K} = \mathbf{Q}_1\mathbf{V}(\mathbf{I} - 2\mathbf{D}^2)^{-1/2}. \tag{14}$$

If $\bar{\mathbf{z}}$ is the vector such that $\bar{z}_i = z_i$ if $w_i \geq 0$ and $-z_i$ otherwise, then substituting from (14), (13) and (11) into (8) and solving gives

$$\hat{\boldsymbol{\beta}} = \mathbf{P}\mathbf{K}^{\mathsf{T}}\sqrt{\bar{\mathbf{W}}}\bar{\mathbf{z}}.$$

The key point about this calculation is that its condition number will be dominated by that of $\mathbf{R}$, the matrix which must be inverted in the definition of $\mathbf{P}$. This is approximately the square root of the condition number for using $\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} + \mathbf{S}$ directly, since the term to be inverted in this latter case would be dominated by $\mathbf{R}^{\mathsf{T}}\mathbf{R}$ (See Golub and van Loan, 1996, sections 2.7.2 and 3.5.4 if this is unclear). The key computational steps involved in finding $\hat{\boldsymbol{\beta}}$ are (9), (10), (12) and (14), plus the rank identification of section 3.3.1.

Given (13), it is now possible to compute one of the REML log determinant components using

$$|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S}| = |\mathbf{R}|^2|\mathbf{I} - 2\mathbf{D}^2|,$$

and it is also worth noting, from (13) and (14), that $(\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S})^{-1} = \mathbf{P}\mathbf{P}^\mathsf{T}$ (strictly some sort of pseudoinverse if there is rank deficiency).

There is an important additional detail. At the penalized MLE, $\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S}$ will be positive semi-definite, so that $d_i \le 1/\sqrt{2}$ (reparameterize so that $\mathbf{R}$ is the identity to see this), but en route to the optimum there is no *guarantee* that the penalized likelihood is positive semi-definite. So, if $d_i > 1/\sqrt{2}$, for any i, then a Fisher step should be substituted. That is set $\alpha_i = 1$, so that $w_i \ge 0 \ \forall i$. Then

$$\mathbf{P} = \mathbf{R}^{-1} \quad \text{and} \quad \mathbf{K} = \mathbf{Q}_1$$

and the expression for $\hat{\boldsymbol{\beta}}$, above, simplifies to $\hat{\boldsymbol{\beta}} = \mathbf{P}\mathbf{K}^\mathsf{T}\sqrt{\mathbf{W}}\mathbf{z}$, while $|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S}| = |\mathbf{R}|^2$.

At the end of model fitting, $\hat{\boldsymbol{\beta}}$ will need to have the pivoting applied at (10) reversed, and the elements of $\hat{\boldsymbol{\beta}}$ that were dropped by the truncation step after (9) will have to be re-inserted as zeroes. Note that the leading order cost of the method described here is the $O(nq^2)$ of the first QR decomposition. LAPACK can be used for all decompositions (Anderson et al. 1999).

3.3.1 λ independent rank deficiency

As mentioned above, it is necessary to deal with any rank deficiency of the weighted penalized least squares problem that is 'structural' to the problem, rather than being the numerical consequence of some smoothing parameter tending to 0 or ∞. That is we need to find which, if any, parameters, $\boldsymbol{\beta}$, would be unidentifiable, even if the penalties and models matrix were all evenly scaled relative to each other.

To achieve this, first find, $\bar{\mathbf{E}}$, a matrix such that

$$\bar{\mathbf{E}}^\mathsf{T}\bar{\mathbf{E}} = \sum_i \mathbf{S}_i/\|\mathbf{S}_i\|_F.$$

The scaling of each component of $\mathbf{S}$ by its Frobenius norm, is simply to achieve even scaling of the components. The required square root can be obtained by symmetric eigen or pivoted Choleski decomposition. Now, using the factor $\boldsymbol{\mathcal{R}}$, from (9), and scaling it by its Frobenius norm, form a pivoted QR decomposition

$$\begin{pmatrix} \boldsymbol{\mathcal{R}}/\|\boldsymbol{\mathcal{R}}\|_F \\ \bar{\mathbf{E}}/\|\bar{\mathbf{E}}\|_F \end{pmatrix} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$$

and determine the rank, r, of the problem from the pivoted triangular factor $\bar{\mathbf{R}}$ (see Cline et al., 1979 and Golub and van Loan, 1996). The pivoting and rank determination indicates which parameters are unidentifiable (e.g. Golub and van Loan, 1996, section 5.5).

3.4 The derivatives of $\hat{\boldsymbol{\beta}}$ with respect to the log smoothing parameters

The preceding Newton based computation of the coefficients, $\hat{\boldsymbol{\beta}}$, leads to some moderately simple expressions for the derivatives of $\hat{\boldsymbol{\beta}}$ with respect to $\rho_j = \log(\lambda_j)$, which will be needed subsequently. Specifically

$$\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j} = -e^{\rho_j}\mathbf{P}\mathbf{P}^\mathsf{T}\mathbf{S}_j\hat{\boldsymbol{\beta}}$$

and

$$\frac{\mathrm{d}^2\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j\mathrm{d}\rho_k} = \delta_j^k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} - \mathbf{P}\mathbf{P}^\mathsf{T}\left\{\mathbf{X}^\mathsf{T}\mathbf{f}^{jk} + e^{\rho_j}\mathbf{S}_j\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_k} + e^{\rho_k}\mathbf{S}_k\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j}\right\}$$

where $\delta_j^k = 1$ if $j = k$ and 0 otherwise, while

$$f_i^{jk} = \frac{1}{2}\frac{\mathrm{d}\eta_i}{\mathrm{d}\rho_j}\frac{\mathrm{d}\eta_i}{\mathrm{d}\rho_k}\frac{\mathrm{d}w_i}{\mathrm{d}\eta_i} \quad \text{and} \quad \frac{\mathrm{d}\boldsymbol{\eta}}{\mathrm{d}\rho_j} = \mathbf{X}\frac{\mathrm{d}\hat{\boldsymbol{\beta}}}{\mathrm{d}\rho_j}.$$

Appendix C provides the derivation of these results, while Appendix D gives the expression for $\mathrm{d}w_i/\mathrm{d}\eta_i$. The leading order cost of these calculations is $O(M^2nq)$ where M is the number of smoothing parameters.

3.5 The rest of the REML objective and its derivatives

Given $\mathrm{d}\hat{\boldsymbol{\beta}}/\mathrm{d}\rho_j$ and $\mathrm{d}^2\hat{\boldsymbol{\beta}}/\mathrm{d}\rho_j\mathrm{d}\rho_k$ then the corresponding derivatives of $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ follow immediately. The derivatives of D w.r.t. $\boldsymbol{\rho}$ are then routine to calculate (see Wood, 2008 for full details). The remaining quantities in the REML (or ML) calculation are $|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}+\mathbf{S}|$, $\hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}\hat{\boldsymbol{\beta}}$ and the log saturated likelihood. These are covered here.

3.5.1 The derivatives of $\log|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}+\mathbf{S}|$

Computation of $\log|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}+\mathbf{S}|$ itself was covered in section 3.3. It will be stable provided computations are conducted in the transformed space. The derivatives are also needed. Defining (with reference to Appendix D)

$$\mathbf{T}_j = \mathrm{diag}\left(\frac{1}{|w_i|}\frac{\partial w_i}{\partial \rho_j}\right) \quad \text{and} \quad \mathbf{T}_{jk} = \mathrm{diag}\left(\frac{1}{|w_i|}\frac{\partial^2 w_i}{\partial \rho_j \partial \rho_k}\right),$$

then some calculations using (16) and (17) from Appendix B show that

$$\frac{\partial \log|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}+\mathbf{S}|}{\partial \rho_k} = \mathrm{tr}\left(\mathbf{K}^\mathsf{T}\mathbf{T}_k\mathbf{K}\right) + e^{\rho_k}\mathrm{tr}\left(\mathbf{P}^\mathsf{T}\mathbf{S}_k\mathbf{P}\right)$$

and

$$\begin{aligned}\frac{\partial^2 \log|\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}+\mathbf{S}|}{\partial \rho_k \partial \rho_j} &= \mathrm{tr}\left(\mathbf{K}^\mathsf{T}\mathbf{T}_{kj}\mathbf{K}\right) + \delta_k^j e^{\rho_k}\mathrm{tr}\left(\mathbf{P}^\mathsf{T}\mathbf{S}_k\mathbf{P}\right) - \mathrm{tr}\left(\mathbf{K}^\mathsf{T}\mathbf{T}_k\mathbf{K}\mathbf{K}^\mathsf{T}\mathbf{T}_j\mathbf{K}\right) \\ &- e^{\rho_j}\mathrm{tr}\left(\mathbf{K}^\mathsf{T}\mathbf{T}_k\mathbf{K}\mathbf{P}^\mathsf{T}\mathbf{S}_j\mathbf{P}\right) - e^{\rho_k}\mathrm{tr}\left(\mathbf{K}^\mathsf{T}\mathbf{T}_j\mathbf{K}\mathbf{P}^\mathsf{T}\mathbf{S}_k\mathbf{P}\right) - e^{\rho_k+\rho_j}\mathrm{tr}\left(\mathbf{P}^\mathsf{T}\mathbf{S}_k\mathbf{P}\mathbf{P}^\mathsf{T}\mathbf{S}_j\mathbf{P}\right).\end{aligned}$$

Although $\mathbf{K}$, $\mathbf{P}$ and the $\mathbf{T}$ matrices all differ from those in Wood (2008), it is none the less possible to employ the tricks laid out in Appendix C of Wood (2008) to efficiently evaluate the various traces in these expressions. The equivalent term for ML is slightly more involved and Appendix E provides details. Note that this step dominates the method's computational cost. The cost of second derivatives is $O(Mnq^2/2)$, while the cost of first derivatives is $O(nq^2)$ (the same as estimating $\boldsymbol{\beta}$). For large M, these costs suggest that quasi-Newton, which only requires first derivatives, will sometimes be more efficient than full Newton for optimization w.r.t. $\boldsymbol{\rho}$, although the fact that quasi-Newton converges more slowly than Newton complicates the comparison.

3.5.2 The derivatives of $\hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}\hat{\boldsymbol{\beta}}$

To complete the derivatives of D_p requires the derivatives of $\hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}\hat{\boldsymbol{\beta}}$. These are readily seen to be

$$\frac{\partial \hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}\hat{\boldsymbol{\beta}}}{\partial \rho_k} = 2\frac{\partial \hat{\boldsymbol{\beta}}^\mathsf{T}}{\partial \rho_k}\mathbf{S}\hat{\boldsymbol{\beta}} + e^{\rho_k}\hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}_k\hat{\boldsymbol{\beta}}$$

and

$$\frac{\partial^2 \hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}\hat{\boldsymbol{\beta}}}{\partial \rho_k \partial \rho_j} = 2\frac{\partial^2 \hat{\boldsymbol{\beta}}^\mathsf{T}}{\partial \rho_k \partial \rho_j}\mathbf{S}\hat{\boldsymbol{\beta}} + 2\frac{\partial \hat{\boldsymbol{\beta}}^\mathsf{T}}{\partial \rho_k}\mathbf{S}_j\hat{\boldsymbol{\beta}}e^{\rho_j} + 2\frac{\partial \hat{\boldsymbol{\beta}}^\mathsf{T}}{\partial \rho_j}\mathbf{S}_k\hat{\boldsymbol{\beta}}e^{\rho_k} + 2\frac{\partial \hat{\boldsymbol{\beta}}^\mathsf{T}}{\partial \rho_k}\mathbf{S}\frac{\partial \hat{\boldsymbol{\beta}}}{\partial \rho_j} + \delta_j^k e^{\rho_k}\hat{\boldsymbol{\beta}}^\mathsf{T}\mathbf{S}_k\hat{\boldsymbol{\beta}},$$

which have $O(M^2q^2)$ computational cost.

3.5.3 Scale parameter related derivatives

For known scale parameter cases, all the derivatives required for direct Newton optimization of the REML or ML criteria have now been obtained. However when ϕ is unknown some further work is still needed (the dependence on ϕ has none of the exploitable linearity of the dependence on λ_i, which is why it must be treated separately).

If $\phi = e^{\rho_\phi}$ is estimated by direct REML then we need only:

$$-\frac{\partial l_r}{\partial \rho_\phi} = -\frac{D_p}{2\phi} - l_s'(\phi)\phi - \frac{M_p}{2}, \qquad -\frac{\partial^2 l_r}{\partial \rho_\phi^2} = \frac{D_p}{2\phi} - l_s''(\phi)\phi^2 - l_s'(\phi)\phi, \qquad -\frac{\partial^2 l_r}{\partial \rho_\phi \partial \rho_k} = -\frac{1}{2\phi}\frac{\partial D_p}{\partial \rho_k}$$

and the derivatives of l_r w.r.t. $\boldsymbol{\rho}$. (These derivatives also serve to emphasize that direct estimation only works with full likelihood, not quasi-likelihood.)

If $\hat{\phi}$ is the Pearson statistic over $n - M_p$, where M_p is the penalty null space dimension (number of fixed effects), then an alternative version of the REML score and its derivatives is as follows:

$$-\hat{l}_r = \frac{D_p}{2\hat{\phi}} - l_s(\hat{\phi}) + K - \frac{M_p}{2}\log(2\pi\hat{\phi}), \qquad -\frac{\partial \hat{l}_r}{\partial \rho_k} = \frac{\partial D_p}{\partial \rho_k}\frac{1}{2\hat{\phi}} - \left(\frac{D_p}{2\hat{\phi}^2} + l_s'(\hat{\phi}) + \frac{M_p}{2\hat{\phi}}\right)\frac{\partial \hat{\phi}}{\partial \rho_k} + \frac{\partial K}{\partial \rho_k},$$

and

$$-\frac{\partial^2 \hat{l}_r}{\partial \rho_k \partial \rho_j} = \frac{\partial^2 D_p}{\partial \rho_k \partial \rho_j}\frac{1}{2\hat{\phi}} - \left(\frac{\partial D_p}{\partial \rho_k}\frac{\partial \hat{\phi}}{\partial \rho_j} + \frac{\partial D_p}{\partial \rho_j}\frac{\partial \hat{\phi}}{\partial \rho_k}\right)\frac{1}{2\hat{\phi}^2} + \left(\frac{D_p}{\hat{\phi}^3} - l_s''(\hat{\phi}) + \frac{M_p}{2\hat{\phi}^2}\right)\frac{\partial \hat{\phi}}{\partial \rho_k}\frac{\partial \hat{\phi}}{\partial \rho_j}$$
$$- \left(\frac{D_p}{2\phi^2} + l_s'(\hat{\phi}) + \frac{M_p}{2\hat{\phi}}\right)\frac{\partial^2 \hat{\phi}}{\partial \rho_k \partial \rho_j} + \frac{\partial^2 K}{\partial \rho_k \partial \rho_j}.$$

These require the derivatives of $\hat{\phi}$, which are easily obtained from the known derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. the smoothing parameters, combined with the derivatives of the Pearson statistic, given in Appendix F.

The ML derivative expressions are identical to those given in this subsection, if one sets $M_p = 0$ (for ML, the fixed effects are not integrated out, and in consequence the direct dependence on the number of fixed effects goes.) Whichever version of REML or ML is used, derivatives of the saturated log likelihood w.r.t. ϕ are required: Appendix G gives some common examples.

3.6 Other smoothness selection criteria

While it was not possible to adapt the Wood (2008) method to reliably optimize REML/ML, the method proposed here can readily optimize prediction error criteria of the sort discussed in Wood (2008). In fact the new method has the advantage of eliminating a potential difficulty with the Wood (2008) method, namely that when using a non-canonical link in the presence of outliers, the Fisher based PIRLS could (rarely) require step length reduction at convergence, which could cause the subsequent derivative iterations to fail.

Prediction error criteria are based on the the deviance, Pearson statistic and effective degrees of freedom of the model, formally defined as $\texttt{tr}(\mathbf{F})$ where $\mathbf{F} = (\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}$. Clearly the methods described so far deal with the deviance and Pearson statistic, but the derivatives of $\texttt{tr}(\mathbf{F})$ require some more work. The results of this are provided in Appendix H. Note that there are good reasons for preferring $\mathbf{W}$ to be based on the Fisher weights in the computation of $\mathbf{F}$. Doing so guarantees that both $\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S}$ and $\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}$ are positive definite, which ensures that the effective degrees of freedom are well defined. There are also robustness-to-outlier arguments (e.g. Demidenko, 2004) for using the Fisher weights for constructing variance estimates, despite the general superiority of observed information over expected information for this purpose (Efron and Hinkley, 1978).

4 Some simulation comparisons

The REML and ML based methods, proposed here, were compared to GCV (AIC for known scale parameters) and PQL (based on the version implemented in R function `glmmPQL`, Venables and Ripley, 2002), as means for selecting smoothing parameters. For each replicate, 400 data, y_i, were simulated (independently) from an exponential family distribution, with mean μ_i where

$$g(\mu_i)/k = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}).$$

g is a known link function and the x_{ji} are i.i.d. uniform on $(0, 1)$. k is used to control the signal to noise ratio. The f_j are plotted at the lower right of figure 2. Five distribution-link combinations were used, with 200 replicates performed for each: normal-identity, gamma-log, Tweedie-log (variance power 1.5), binary-logit and Poisson-log. For each case k was set to achieve a squared correlation coefficient between μ_i and y_i of about 0.5. A generalized additive model with the correct link-error structure was fitted to each replicate, but with the linear predictor given by a sum of smooth functions of the 3 actual predictors plus a smooth function of a nuisance predictor, which was i.i.d. uniform, but did not influence the true μ_i. The 4 component smooth models were represented by rank 10 thin plate regression splines (Wood, 2003), except for the 3rd component, for which a rank of 30 was used. Smoothing parameters were chosen by each of REML, ML, PQL and GCV (or AIC when the scale parameter was known), for each replicate.

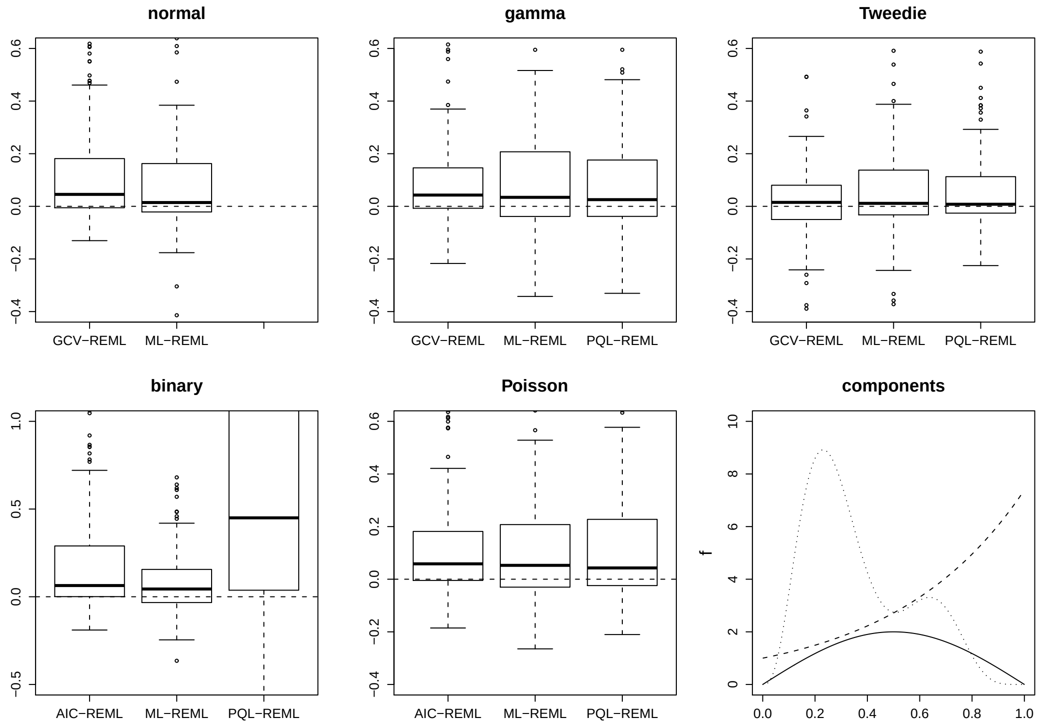

Figure 2: Mean Square Error comparisons between REML and other methods for 5 distributions. Data were simulated using an additive linear predictor, made up from the 3 functions shown in the lower right panel. The linear predictor was scaled so that there was about 50% unexplained variance in each replicate dataset. Generalized additive models were fitted to each replicate dataset, using the correct distribution and link, with smoothing parameters chosen by REML, ML, GCV (AIC for known ϕ) or PQL. Boxplots show the distributions, over 200 replicates, of differences in mean square error between each alternative method and REML. MSE is measured on the scale of the linear predictor, for all distributions except binary, where it is on the probability scale. Prior to plotting, the MSE are divided by the MSE for REML estimation, averaged over the the case being plotted. In all cases a Wilcoxon signed rank test indicates that REML has lower MSE than the competing method (p value $< 10^{-3}$ except for the PQL-ML comparison for the Tweedie, where p=0.04). The Tweedie variance power was 1.5. The log link was used for all cases except normal (identity) and binary (logit). PQL failed in 16, 10, 22 and 7 replicates, for gamma, Tweedie, binary and Poisson data respectively. The other methods converged successfully for every replicate. PQL was between 10 and 20 times slower than the alternatives. See section 4.

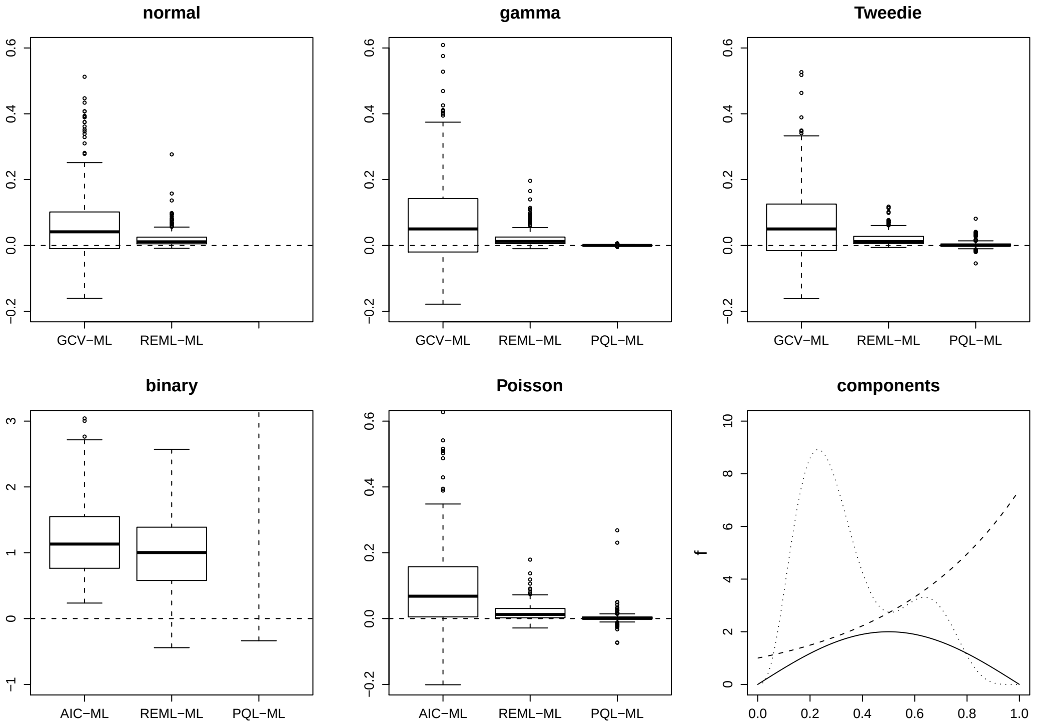

Figure 3: As figure 2, but using data for which only 5% of the variance in the response was noise. In this case ML gave the best MSE performance, so has replaced REML as the reference method. All differences are significant at $p < 0.00004$ except the PQL-ML comparisons for Gamma, Tweedie and Poisson for which the p-values are 0.01, 0.01 and 0.0006. See section 4.

Model performance was judged by calculating the mean square error in reconstructing the true linear predictor, at the observed covariate values. In the case of binary data, this measure is rather unstable for fitted probabilities in the vicinity of 0 or 1, so the probability scale was used in place of the linear predictor scale.

The results are summarized in Figure 2. In all cases REML gave better reconstructions, with Wilcoxon signed rank tests usually strongly suggesting that the median difference was not zero, and REML giving the lower MSE error on average. The most dramatic difference is between REML and PQL for binary data, where PQL has a substantial tail of poor fits, reflecting the well known fact that PQL is poor for binary data. Notice also the skew in the GCV-REML comparisons: this seems to result from a smallish proportion of GCV/AIC based replicates substantially over-fitting.

GCV/AIC, REML and ML fits did not fail for these examples, while PQL had a 4-10% failure rate. Mean time per replicate for GCV/AIC, REML and ML was about 0.7 seconds on a 1.33GHz Intel U7700 running linux (mid-range laptop). PQL was between 10 and 20 times slower. All computations were performed with R 2.9.2 (R core development team, 2008) and R package `mgcv` version 1.6-1 (which includes a Tweedie family based on Dunn and Smith, 2005).

The experiment was repeated at lower noise levels. First for noise levels such that the r^2 between μ_i and y_i was about 0.7 and then for still lower noise levels so that the r^2 was about 0.95. Figure 3 shows the results for the lowest noise level. In this case ML gives the best MSE performance, although REML is not much worse and still better than the prediction error criteria. The intermediate noise level results are not shown, but show ML and REML to be almost indistinguishable, and both better than prediction error criteria. It seems likely that the superiority of ML over REML in the lowest noise case relates to Wahba's (1985) demonstration that REML undersmooths, asymptotically: ML will of course smooth more, but is still consistent (Kauermann et al. 2009). Similarly the failure of prediction error methods to show any appreciable catch up as noise levels were reduced, despite their asymptotic superiority in MSE terms, presumably relates to the excruciatingly slow convergence rates for prediction criteria based estimates, obtained in Härdle, Hall and Marron (1988).

The two problematic examples from the introduction to Wood (2008, see figures 1 and 2) were also repeated with the methods developed here: convergence was unproblematic and reasonable fits were obtained. See appendix A for some further comparisons with another alternative method.

The simulation evidence supports the implication of Reiss and Ogden's (2009) work, that REML (and hence the structurally very similar ML) may have practical advantages over GCV/AIC for smoothing parameter selection, and reinforces the message from Wood (2008), that direct nested optimization is quicker and more reliable than selecting smoothing parameters based on approximate working models.

5 Examples

This section presents 3 example applications which, as special cases of penalized GLMs, are straightforward given the general method proposed in this paper.

5.1 Simple P-spline adaptive smoothing

An important feature of the proposed method is that it is stable even when different penalties act on intersecting sets of parameters. Tensor product smooths used for smooth interaction terms are an obvious important case where this occurs (see e.g. Wood, 2006 section 4.1.8), but adaptive smoothing provides a less well known example, as illustrated in this section, using adaptive P-splines.

The 'P-splines' of Eilers and Marx (1996) combine B-spline basis functions and discrete penalties on the basis coefficients, to obtain flexible spline like smoothers. For example, if we let $b_j(x)$ denote B-spline basis functions, with evenly spaced knots, then an unknown function f can be represented (approximately) as

$$f(x) = \sum_{i=1}^{K} \beta_i b_i(x)$$

and the wiggliness of this function can be measured using the discrete penalty

$$\mathcal{P}_{\text{ordinary}} = \sum_{i=2}^{K-1} (\beta_{i-1} - 2\beta_i + \beta_{i+1})^2,$$

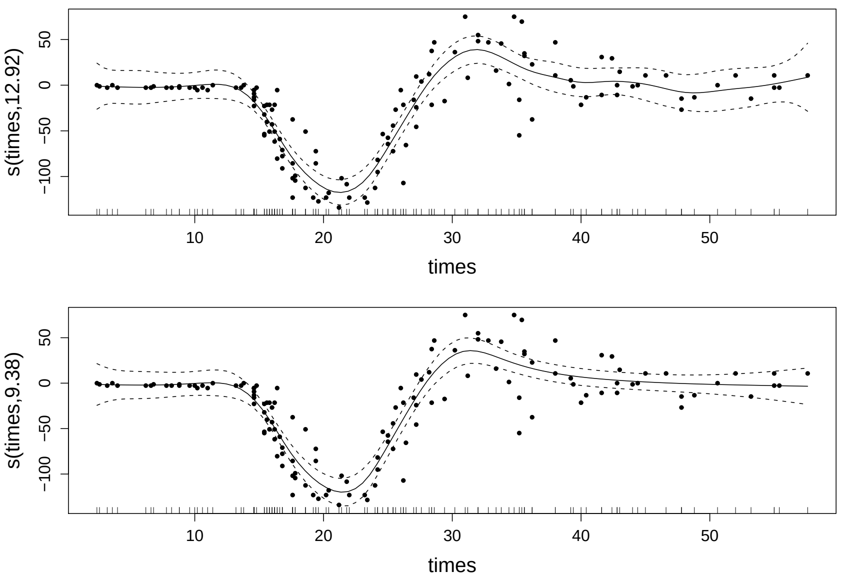

Figure 4: Two attempts to smooth the motorcycle crash data. The top panel represents the smooth as a rank 40 penalized thin plate regression spline, while the lower panel uses a simple adaptive smoother of the type discussed in Section 5.1. All smoothing parameters were chosen by REML. Notice that the adaptive smoother uses fewer effective degrees of freedom and produces a fit which appears to show better local adaptation to the data.

or higher or lower order alternatives. The penalty can be used as a smoothing penalty in fitting. One of the reasons that P-splines have proved so popular, is the ease with which they can be modified to perform non-standard smoothing tasks, at relatively little loss of performance relative to more computationally complex smoothers. Adaptive smoothing illustrates this.

An adaptive penalty is easily constructed by allowing the terms in the penalty to have different weights, depending on on i, and hence on x. For example:

$$\mathcal{P} = \sum_{i=2}^{K-1} c_i(\beta_{i-1} - 2\beta_i + \beta_{i+1})^2.$$

Now defining $d_i = \beta_{i-1} - 2\beta_i + \beta_{i+1}$, and $\mathbf{D}$ to be the matrix of coefficients such that $\mathbf{d} = \mathbf{D}\boldsymbol{\beta}$, we have $\mathcal{P} = \boldsymbol{\beta}^{\mathsf{T}}\mathbf{D}^{\mathsf{T}}\text{diag}(\mathbf{c})\mathbf{D}\boldsymbol{\beta}$. The elements, c_i, are unknown, but we could use a B-spline basis to model the c_i as a smooth function of i or x so that $\mathbf{c} = \mathbf{C}\boldsymbol{\lambda}$, where λ is a vector of unknown (positive) coefficients. In this case

$$\mathcal{P} = \sum_j \lambda_j \boldsymbol{\beta}^{\mathsf{T}}\mathbf{D}^{\mathsf{T}}\text{diag}(\mathbf{C}_{\cdot,j})\mathbf{D}\boldsymbol{\beta}$$

where $\mathbf{C}_{\cdot,j}$ is column j of $\mathbf{C}$. i.e. the adaptive penalty has become a sum of penalties multiplied by smoothing parameters (λ_j). The same construction can be used for smooths of several covariates, using tensor products of P-splines. See Krivobokova et al. (2008) for a more sophisticated P-spline based approach to this problem.

The obvious advantage of the approach given here is that it allows adaptive smoothers to be used as components of penalized GLMs in the same way as any other smooth. As an example consider smoothing the well known motorcycle crash data used in Silverman (1985). The response, a_i, is acceleration of the head of a test dummy in a simulated motorcycle crash, and it depends on time, t_i. A simple model is

$$a_i = f(t_i) + \epsilon_i$$

where the ϵ_i are i.i.d. $N(0, \sigma^2)$ (although a better model would have σ^2 depending on time as well). Given that the data show a low acceleration phase followed by rapid changes in acceleration followed by smooth return to zero, it is possible to make the case that the degree of penalization of f should depend on t. A model was therefore fitted in which f was represented using a rank 40 cubic B-spline basis (even knot spacing), penalized using the adaptive penalty given above, $\boldsymbol{\lambda}$ having dimension 5 (although the results are rather insensitive to the exact choice here). The smoothing parameters, $\boldsymbol{\lambda}$, were chosen by REML.

The results are shown in Figure 4, which also includes a fit in which a single penalty rank 40 thin plate regression spline is used to represent $f(t)$. The single penalty case has to use the same degree of penalization for all t, with the result that the curve at low and high times appears under-penalized and too bumpy, presumably to accomodate the high degree of variability at intermediate times. The adaptive fit took 1.3 seconds, compared to 0.15 seconds for the single penalty fit (see section 4, for computer details).

5.2 Generalized regression of scalars on functions

The fact that the method described in this paper has been developed for the rather general model (1) means that it can be used for models that superficially appear to be rather different to a GAM. To illustrate this, this section revisits an example from Reiss and Ogden (2009), but makes use of the new method to employ a more general model than theirs, based on non-Gaussian errors with multiple penalties.

Consider a response, y_i, dependent on predictor function, $z_i(x)$, where x may be univariate or multivariate. In this case an appropriate model might be

$$g(\mu_i) = \alpha + \int f(x) z_i(x) dx, \tag{15}$$

with y_i an observation from some exponential family distribution, with mean μ_i. $f(x)$ is an unknown 'coefficient' function, and must be estimated. It is straightforward to extend the model by adding other smooth terms to the linear predictor (right hand side). In practice the integral will be approximated by quadrature, with the midpoint rule being adequate in most cases. Suppose that the domain of $z_i(x)$ is finite and let x_j denote points at which z_i has been observed (with even spacing h). The model becomes

$$g(\mu_i) = \alpha + h \sum_j f(x_j) z_i(x_j).$$

Any penalized regression spline basis can be used for f, and model estimation proceeds as for any other penalized GLM. For more detail on such models see Marx and Eilers (1999); Escabias, Aguilera and Valderrama (2004); Ramsay and Silverman (2005) or Reiss and Ogden (2007) (also Wahba, 1990).

As an example, consider trying to predict the octane rating of gasoline/petrol from its near infra red spectrum. For internal combustion engines in which a fuel air mixture is compressed within the cylinders prior to combustion, it is important that the fuel air mixture does not spontaneously ignite due to compressive heating. Such early combustion results in 'knocking' and poor engine performance. The octane rating of fuel measures its resistance to knocking. It is a somewhat indirect measure: the lowest compression ratio at which the fuel causes knocking is recorded. The octane rating is the percentage of iso-octane in the mixture of n-heptane and iso-octane with the same lowest knocking compression as the fuel sample. Measuring of octane rating requires special variable compression test engines, and it would be rather simpler to measure the octane from spectral measurements on a fuel sample, if this were possible.

The upper left panel of figure 5 shows Near Infra Red (NIR) spectra for 60 gasoline samples (from Kalivas, 1997 as provided by Wehrens and Mevik, 2007). The Octane rating of each sample has also been measured. Model (15) is a possibility for such data (where y_i is octane rating, $z_i(x)$ is the i^{th} spectrum and x is wavelength). The octane rating is positive and continuous (at least in theory), and there is some indication of increasing variance with mean (see figure 5c), so a gamma distribution with log link is an appropriate initial model. The spectra themselves are rather spiky, with some smooth regions interspersed with regions of very rapid variation. It seems sensible to allow the coefficient function, $f(x)$, the possibility of behaving in a similar way, so representing f using the same sort of adaptive smooth used in the previous section is appropriate. Estimation of this model is then just a case of estimating a GLM subject to multiple penalization. The remaining panels of figure 5 show the results of this fitting, with REML smoothness selection.

Notice that the coefficient function appears to be contrasting the two peak regions with the trough between them, with the extreme ends of the spectra apparently adding little. The model explains around 98% of the deviance in octane rating, and the residual plots look plausible (including QQ-plot of deviance residuals, not shown).

5.3 GAM term selection and null space penalties

Smoothing parameter selection does most of the work in selecting between models of differing complexity, but does not usually remove a term from the model altogether. If the smoothing parameter for a term tends to infinity, this

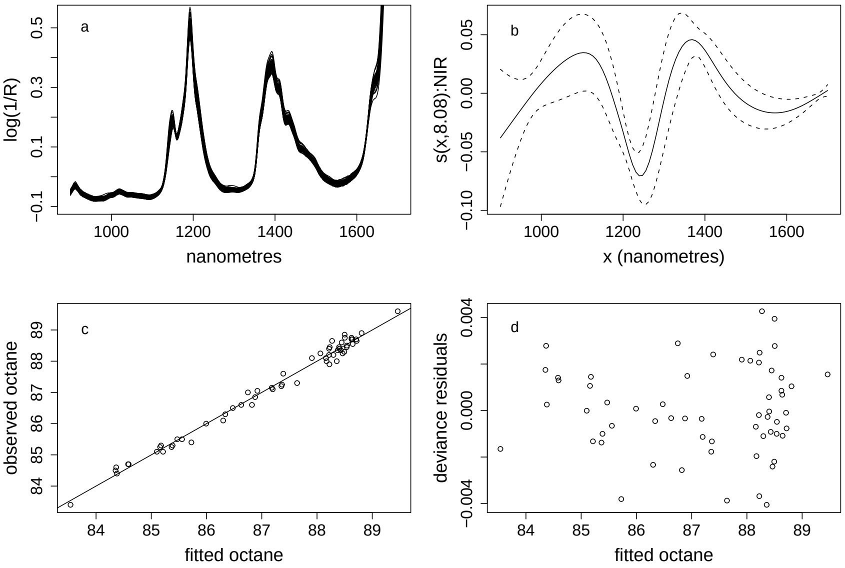

Figure 5: a) Near Infra Red (NIR) spectra for 60 samples of gasoline. The y axis is the log of the inverse of reflectance, which is measured every 2 nanometres. These spectra ought to be able to predict the octane rating of the samples. The spectra actually reach 1.2 at the right hand end, but since this region turns out to have little predictive power, the y axis has been truncated, in order to show more detail at lower wavelengths. b) The estimated coefficient function for the octane — NIR model given in section 5.2 (with factor h absorbed): the inner product of this with the spectrum for a sample gives the predicted octane rating. c) The observed octane ratings against the fitted. d) The deviance residuals for the model, against fitted octane rating. See section 5.2.

usually causes the term to tend towards some simple, but non-zero, function of its covariate. For example, as its smoothing parameter tends to infinity, a cubic regression spline term will tend to a straight line. It seems logical to decide on whether or not terms should be included in the model using the same criterion used for smoothness selection, but how should this be achieved in practice? Tutz and Binder (2006) proposed one solution to the model selection problem, by using a boosting approach to perform fitting, smoothness selection and term selection simultaneously. They also provide evidence that in very data poor settings, with many spurious covariates, this approach can be much better than the alternatives. This section proposes a possible alternative to boosting, in which each smooth term is given an extra penalty, which will shrink to zero functions that are in the null space of the usual penalty.

For example, consider a smooth with K coefficients, $\boldsymbol{\beta}$, and penalty matrix $\mathcal{S}$, with null space dimension M_s, so that the wiggliness penalty is $\boldsymbol{\beta}^{\mathsf{T}}\mathcal{S}\boldsymbol{\beta}$. Now consider the eigen-decomposition $\mathcal{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathsf{T}}$. The first $K - M_s$ eigenvalues Λ_i will be positive, and the last M_s will be zero. Writing $\boldsymbol{\Lambda}_+$ for the $(K - M_s) \times (K - M_s)$ diagonal matrix containing only the positive eigenvalues, and $\mathbf{U}_+$ for the $K \times (K - M_s)$ matrix of corresponding eigenvectors, then $\mathcal{S} = \mathbf{U}_+\boldsymbol{\Lambda}_+\mathbf{U}_+^{\mathsf{T}}$. Now let $\mathbf{U}_-$ be the $K \times M_s$ matrix of the eigenvectors corresponding to zero eigenvalues. $\mathbf{U}_+$ forms a basis for the space of coefficients corresponding to the 'wiggly' component of the smooth, while $\mathbf{U}_-$ is a basis for the components of zero wiggliness — the null space of the penalty. The two bases are orthogonal. So, if we want to produce a penalty which penalizes only the null space of the penalty, we could use $\boldsymbol{\beta}^{\mathsf{T}}\mathcal{S}_N\boldsymbol{\beta}$ where $\mathcal{S}_N = \mathbf{U}_-\mathbf{U}_-^{\mathsf{T}}$. If a smooth term is already subject to multiple penalties (e.g. a tensor product smooth or an adaptive smooth), the same basic construction holds, but the null space is obtained from the eigen decomposition of the sum of the original penalty matrices. Notice that this construction is general and completely automatic.

This sort of construction could be used with any smoothing parameter selection method, not just RE/ML, but it is less appealing if used with a method which is prone to undersmoothing, as GCV seems to be.

As a small example, Poisson data were simulated assuming a log link and a linear predictor made up of the sum of the 3 functions shown at the lower right of Figure 2, applied to 3 sets of 200 i.i.d. $U(0, 1)$ covariates. 6 more i.i.d. $U(0, 1)$ nuisance covariates were simulated. A GAM was fitted to the simulated data, assuming a Poisson distribution and log link, and with a linear predictor consisting of a sum of 9 smooth functions of the 9 covariates. Each smooth function was represented using a rank 10 cubic regression spline (actually P-splines for GAMboost). The model was fitted using 4 different methods: the GAM boosting method of Tutz and Binder, using version 1.1 of R package

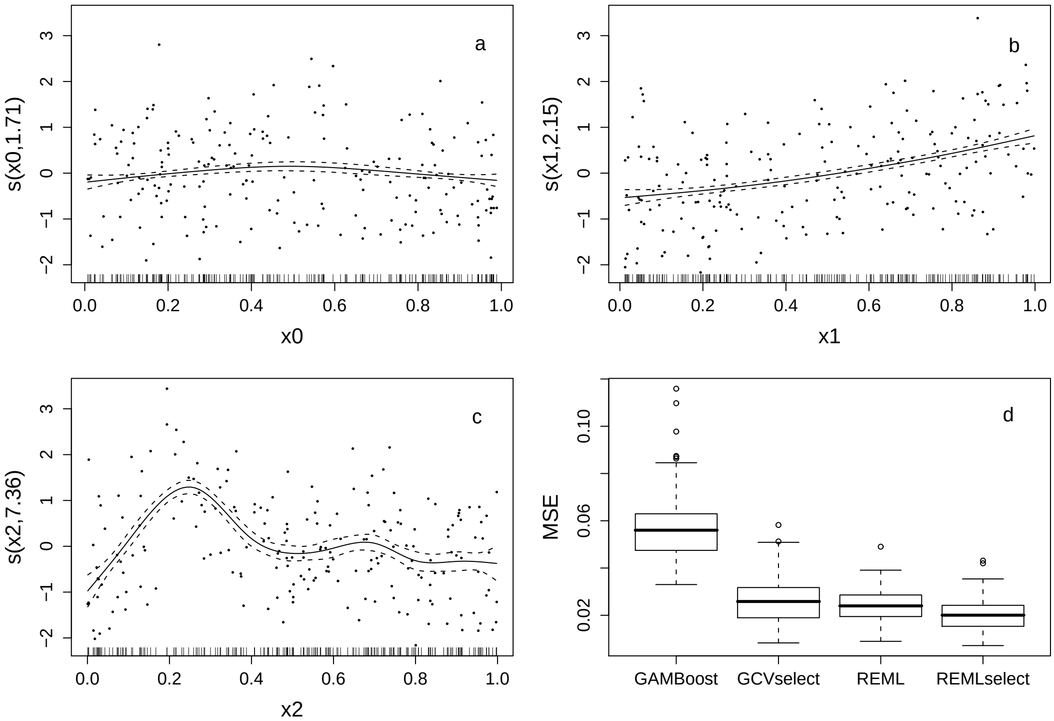

Figure 6: Model selection example from section 5.3. Models were fitted to Poisson data simulated from a linear predictor made up of the 3 terms shown in the lower right panel of Figure 2. The linear predictors of the fitted models also included smooth functions of 6 additional nuisance predictors. 4 alternative fitting methods were used for each replicate simulation. a) - c) show typical estimates of the terms that actually made up the true linear predictor (using REML, with selection penalties). Partial Pearson residuals are shown for each smooth estimate. d) shows the distribution, over 200 replicates, of the mean square error of the models fitted by each of the methods. 'GAMBoost' is fitted using Tutz and Binder's (2006) boosting method, 'GCVselect' is for models with selection penalties under GCV smoothness selection, 'REML' is REML smoothness selection without selection penalties, while 'REMLselect' is for REML smoothness selection, with selection penalties.

GAMBoost (with penalty set to 500 to ensure that each fit used well over the 50 boosting steps suggested as the minimum by Tutz and Binder); GCV smoothness selection, with the null space penalties suggested here, REML with no null space penalties and REML with null space penalties. 200 replicates of this experiment were run, and the mean square error in the linear predictor at the covariate values was recorded for each method for each replicate.

Figure 6 shows the results. REML with null space penalties achieves lower MSE than REML without null space penalties, and substantially better performance than GCV with null space penalties or GAM boosting. The success of the methods in identifying which components should be in the model at all was also recorded. For GAMBoost the methods given in the GAMBoost package were employed, while for the null space penalties, terms with effective degrees of freedom greater than 0.2 were deemed to have been selected. On this basis the false negative rates (rates at which influential covariates were not selected) were .6% for boosting and .16% for the other methods. The false positive rates (rates at which spurious terms were selected) were 67%, 71% and 62% for boosting, GCV and REML, respectively. REML with null space penalties took just under 6 seconds per fit, on average, while boosting took about 2.5 minutes per fit. Note that the example here has relatively high information content, relative to the scenarios investigated by Binder and Tutz: with less information boosting is still appealing.

6 Discussion

The method proposed in this paper offers a general computationally efficient way of estimating the smoothing parameters of models of the form (1), when the f_j are represented using penalized regression splines and the coefficients, $\boldsymbol{\beta}$, are estimated by optimizing (3). With this method, REML/ML based estimation of semi parametric GLMs can rival the estimation of ordinary parametric GLMs for routine computational reliability. Previously such efficiency and reliability were only available for prediction error criteria, such as GCV. This means that the advantages of REML/ML estimation outlined in section 1.1 need no longer be balanced against the more reliable fitting methods available for GCV/AIC. The cost of this enhancement is that the proposed method has a somewhat more complex mathematical structure than the previous prediction error based methods (e.g. Wood, 2008), but since the method is freely available in R package mgcv (from version 1.5) this is not an obstacle to its use.

Given that RE/ML estimation requires that we view model (1) as a generalized linear mixed model, then an obvious question is why it should be treated as a special case for estimation purposes, rather than estimated by general GLMM software? The answer lies in the special nature of the λ_i. The fact that they enter the penalty/precision matrix linearly, facilitates both the evaluation of derivatives to computational accuracy, and the ability to stabilize the computations via the method of Appendix B. In addition the λ_i are unusual precision parameters in that their 'true' value is often infinite. The latter behaviour can cause problems for general purpose methods, which can not exploit the advantages of the linear structure. Conversely, the method proposed here can be used to fit any GLMM where the precision matrix is a linear combination of known matrices, but since it is not designed to exploit the the sparse structure that many random effects have, it may not be the most efficient method for so doing.

A limitation of the method presented here is that it is designed to be efficient when the f_j are represented using penalized regression splines as described in Wahba (1980), Parker and Rice (1985), Eilers and Marx (1996), Marx and Eilers (1998), Ruppert, Wand and Carroll (2003), Wood (2003) etc. These 'intermediate rank' smoothers have become very popular over the last decade, as researchers realized that many of the advantages of splines could be obtained without the computational expense of full splines: an opinion which turns out to be well founded theoretically (see Gu and Kim, 2002; Hall and Opsomer, 2005; Kauermann, Krivobokova and Fahrmeir, 2009). But despite its wide applicability, the penalized regression spline approach has limitations. The most obvious is that relatively low rank smooths are unsuitable for modelling short range autocorrelation (particularly spatial). Where this deficiency matters, Rue et al. (2008) offer an attractive alternative approach, by directly estimating additive smooth components of the linear predictor, with very sparse $\mathbf{S}_j$ matrices directly penalizing these components. The required sparsity can be obtained by modelling the smooth components as Markov Random Fields of some sort. Provided that the number of smoothing parameters is quite low, then the methods offer very efficient computation for this problem class, as well as better inferences about the smoothing parameters themselves. When the model includes large numbers of random effects, but not all components have the sparsity required by Rue et al., or when the number of smoothing parameters/ variance parameters is moderate to large, then the simulation based Bayesian approach of Fahrmeir, Lang and co-workers (e.g. Lang and Brezger, 2004, Brezger and Lang, 2006, Fahrmeir and Lang, 2001) is likely to be more efficient than the method proposed here, albeit applicable to a more restricted range of penalized GLMs, because of restrictions on the $\mathbf{S}_j$ required to maintain computational efficiency.

An interesting area for further work would be to establish relative convergence rates for the $\hat{f}_j$ under REML, ML and GCV smoothness selection. It is not hard to arrange for $\hat{f}_j$ to be consistent under either approach, at least when spline like bases are used for the f_j in (1). Without penalization, all that we require is that the basis dimensions grows with sample size, n, fast enough that the spline approximation error declines at a faster rate than the sampling variance of $\hat{f}_j$, but slow enough that $\dim(\boldsymbol{\beta})/n \to 0$ (so that the observed likelihood converges to its expectation). This is not difficult to achieve, given the good approximation theoretic properties of splines. If smoothing parameters are chosen to be small enough, then penalization will *reduce* the MSE at any n, so consistency can be maintained under penalized estimation. In fact, asymptotically, GCV *minimizes* MSE (or a generalized equivalent), so the $\hat{f}_j$ will be consistent under GCV estimation. Since REML smooths less than GCV, asymptotically (Wahba, 1985), then the same must hold for REML. However, establishing the relative convergence *rates* actually achieved under the two alternatives appears to be more involved.

Acknowledgements

I am especially grateful to Mark Bravington for explanation of the implicit function theorem, and for providing the original Fisheries modelling examples that broke Fisher scoring based approaches, and to Phil Reiss for first alerting me to the real practical advantages of REML and providing an early pre-print of Reiss and Ogden (2009). CSIRO paid for a visit to Hobart where the Tweedie work was done, and it became clear that the Wood (2008) method could not be extended to REML. Thanks also to Mark Bravington, Merrilee Hurn and Alistair Spence for a number of helpful discussions during the painful process that lead to the Section 3.1/Appendix B method. Phil Reiss, Mark Bravington, Nicole Augustin and Giampiero Marra all provided very helpful comments on an earlier draft of this paper. The editor, associate editor and two anonymous referees all made suggestions which substantially improved the paper, for which I am also grateful.

Appendix A: Convergence failures of previous REML schemes

Wood (2008) provides a number of examples of convergence failure for the PQL approach, in which smoothing parameters are estimated iteratively by RE/ML estimation of working linear mixed models. The alternative scheme proposed in the literature is implemented by Brezger, Kneib and Lang (2007) in the BayesX package. Like PQL, this scheme need not converge (as Brezger et al. explicitly point out), but Brezger et al. employ an ingenious heuristic stabilization trick which seems to lead to superior performance to PQL in this regard. However it is not hard to find realistic examples that still give convergence problems. For example the following code was used in R 2.7.1 to generate data with a relatively benign co-linearity problem and a mild mean variance relationship problem:

```
set.seed(1);n<-1000;alpha <- .75
x0 <- runif(n);x1 <- x0 * alpha + (1-alpha)*runif(n)
x2 <- runif(n);x3 <- x2 * alpha + (1-alpha)*runif(n)
x4 <- runif(n);x5 <- runif(n)
f0 <- function(x) 2 * sin(pi * x)
f1 <- function(x) exp(2 * x)
f2 <- function(x) 0.2*x^11*(10 *(1 - x))^6 + 10*(10*x)^3*(1 - x)^10
f <- f0(x0) + f1(x1) + f2(x2)
y <- rgamma(f,exp(f/4),scale=1.2)
```

Fitting the model

$$\log\{\mathbb{E}(y_i)\} = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + f_4(x_{4i}) + f_5(x_{5i}) + f_6(x_{6i})$$

$y_i \sim$ Gamma, in BayesX 1.5.0, representing each f by a (default) rank 20 P-spline, resulted in convergence failure, with the estimates zig-zagging without ever converging. 9 subsequent replicates of this experiment yielded 2 more convergence failures of the same sort, 3 catastrophic divergences, and 4 problem free convergences (although one of these took more than 200 iterations). Fitting the same model to these data sets using the methods proposed in this paper gave no problems and sensible function reconstructions in each case.

Appendix B: $|\sum_i \lambda_i \mathbf{S}_i|_+$

As discussed in section 3.1, a stable method for calculating $\log|\sum_i \lambda_i \mathbf{S}_i|_+$ and its derivatives w.r.t. $\rho_i = \log \lambda_i$, is required, when the λ_i may be wildly different in magnitude. This appendix provides such a method by extending the simple approach described in section 3.1.

Here it is assumed that $q \times q$ matrix $\mathbf{S} = \sum_i \lambda_i \mathbf{S}_i$ is formally of full rank. When this is not the case then the following initial transformation will be required. First form the symmetric eigen-decomposition:

$$\tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^{\mathsf{T}} = \sum_i \mathbf{S}_i / ||\mathbf{S}_i||_F,$$

where $\|\cdot\|_F$ is the Frobenius norm. Now let $\mathbf{U}_+$ denote the columns of $\tilde{\mathbf{U}}$ corresponding to positive eigenvalues. The transformation $\tilde{\mathbf{S}}_i = \mathbf{U}_+^{\mathsf{T}} \mathbf{S}_i \mathbf{U}_+$ is then applied and the methods of this appendix are utilized on the transformed matrices. It is easy to show that $|\mathbf{S}|_+ = |\sum_i \lambda_i \tilde{\mathbf{S}}_i|$, and that $\sum_i \lambda_i \tilde{\mathbf{S}}_i$ has full rank. For the rest of this appendix it is assumed that this transformation has been applied if necessary, and the tildes are dropped.

Initialization: Set $K = 0$, $Q = q$ and $\bar{\mathbf{S}}_i = \mathbf{S}_i \; \forall \; i$. Set $\gamma = \{1 \ldots M\}$, where M is the number of $\mathbf{S}_i$ matrices.

Similarity transformation: The following steps are iterated until the termination criteria is met (at step 4).

1. Set $\Omega_i = \|\bar{\mathbf{S}}_i\|_F \lambda_i \; \forall \; i \in \gamma$.
2. Create $\alpha = \{i : \Omega_i \geq \epsilon \max(\Omega_i), i \in \gamma\}$ and $\gamma' = \{i : \Omega_i < \epsilon \max(\Omega_i), i \in \gamma\}$ where ϵ is e.g. the cube root of the machine precision. So α indexes the dominant terms out of those remaining.
3. Find the eigenvalues of $\sum_{i \in \alpha} \bar{\mathbf{S}}_i / \|\bar{\mathbf{S}}_i\|_F$ and use these to determine the formal rank, r, of any summation of the form $\sum_{i \in \alpha} \lambda_i \bar{\mathbf{S}}_i$ where the λ_i are positive. Rank is determined by counting the number of eigenvalues that are larger than ε times the dominant eigenvalue. ε is typically the machine precision raised to a power in [0.7,0.9].
4. If $r = Q$ then terminate. The current $\mathbf{S}$ is the one to use for determinant calculation.
5. Find the eigen decomposition $\mathbf{U}\mathbf{D}\mathbf{U}^{\mathsf{T}} = \sum_{i \in \alpha} \lambda_i \bar{\mathbf{S}}_i$, where the eigen-values are arranged in descending order on the leading diagonal of $\mathbf{D}$. Let $\mathbf{U}_r$ be the first r columns of $\mathbf{U}$ and $\mathbf{U}_n$ the remaining columns.
6. Write $\mathbf{S}$ in partitioned form

   $$\mathbf{S} = \begin{pmatrix} \mathbf{A}_{K \times K} & \mathbf{B}_{K \times Q} \\ \mathbf{B}^{\mathsf{T}}_{Q \times K} & \mathbf{C}_{Q \times Q} \end{pmatrix}$$

   where the subscripts in the above denote dimensions (rows × columns). Then set $\mathbf{B}' = \mathbf{B}\mathbf{U}$ and

   $$\mathbf{C}' = \begin{pmatrix} \mathbf{D}_r + \mathbf{U}_r^{\mathsf{T}} \mathbf{S}_{\gamma'} \mathbf{U}_r & \mathbf{U}_r^{\mathsf{T}} \mathbf{S}_{\gamma'} \mathbf{U}_n \\ \mathbf{U}_n^{\mathsf{T}} \mathbf{S}_{\gamma'} \mathbf{U}_r & \mathbf{U}_n^{\mathsf{T}} \mathbf{S}_{\gamma'} \mathbf{U}_n \end{pmatrix}$$

   where $\mathbf{S}_{\gamma'} = \sum_{i \in \gamma'} \lambda_i \bar{\mathbf{S}}_i$. Then

   $$\mathbf{S}' = \begin{pmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{\mathsf{T}} \end{pmatrix} \mathbf{S} \begin{pmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B}' \\ \mathbf{B}'^{\mathsf{T}} & \mathbf{C}' \end{pmatrix}$$

   and $|\mathbf{S}| = |\mathbf{S}'|$. The key point here is that the effect of the terms indexed by α has been concentrated into an $r \times r$ block, with rows and columns to the lower right of that block uncontaminated by 'large machine zeroes' from the terms indexed by α.
7. Define

   $$\mathbf{T}_\alpha = \begin{pmatrix} \mathbf{I}_K & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_r & \mathbf{0} \end{pmatrix} \quad \text{and} \quad \mathbf{T}_{\gamma'} = \begin{pmatrix} \mathbf{I}_K & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{pmatrix}.$$

   and transform

   $$\mathbf{S}_i \leftarrow \mathbf{T}_\alpha^{\mathsf{T}} \mathbf{S}_i \mathbf{T}_\alpha \;\; \forall \;\; i \in \alpha$$

   and

   $$\mathbf{S}_i \leftarrow \mathbf{T}_{\gamma'}^{\mathsf{T}} \mathbf{S}_i \mathbf{T}_{\gamma'} \;\; \forall \;\; i \in \gamma'.$$

   These transformations facilitate derivative calculations using the transformed $\mathbf{S}$.

8. Transform $\bar{\mathbf{S}}_i \leftarrow \mathbf{U}_n^{\mathsf{T}} \bar{\mathbf{S}}_i \mathbf{U}_n \; \forall \; i \in \gamma'$.

9. Set $K \leftarrow K + r$, $Q \leftarrow Q - r$, $\mathbf{S} \leftarrow \mathbf{S}'$ and $\gamma \leftarrow \gamma'$. Return to step 1.

Note that the orthogonal matrix which similarity transforms the original $\mathbf{S}$ to the final transformed version can be accumulated as the algorithm progresses, to produce the $\mathbf{Q}_s$ of section 3.1.

The effect of the preceding iteration is to concentrate the dominant terms in $\mathbf{S}$ into the smallest possible block of leftmost columns, with these terms having no effect beyond those columns. Next the most dominant terms in the remainder are concentrated in the smallest possible number of immediately succeeding columns, again with no effect to the right of these columns. This pattern is repeated. Since QR decomposition operates on columns of $\mathbf{S}$, without mixing columns, it can now be used to stably evaluate the determinant of the transformed $\mathbf{S}$. Alternative methods of determinant calculation (e.g. Choleski or symmetric eigen) would require an additional pre-conditioning step.

It is straightforward to obtain a stable matrix square root of the transformed $\mathbf{S}$, which maintains the column separation evident in $\mathbf{S}$ itself. Defining diagonal matrix $P_{ii} = |S_{ii}|^{1/2}$, form the Choleski factor of the diagonally preconditioned version of $\mathbf{S}$. i.e.

$$\mathbf{L}\mathbf{L}^{\mathsf{T}} = \mathbf{P}^{-1}\mathbf{S}\mathbf{P}^{-1}.$$

Then $\mathbf{E} = \mathbf{L}^{\mathsf{T}}\mathbf{P}$, is a matrix square root, such that $\mathbf{E}^{\mathsf{T}}\mathbf{E} = \mathbf{S}$. Pre-conditioning is essential in order to ensure that the square root is computable without ever requiring numerical truncation, since the latter would cause spurious discontinuous changes in the numerical value of $|\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X} + \mathbf{S}|$, which depends on $\mathbf{E}$.

Finally, note that, based on the general results,

$$\frac{\partial \log |\mathbf{F}|}{\partial x_j} = \text{tr}\left(\mathbf{F}^{-1}\frac{\partial \mathbf{F}}{\partial x_j}\right) \tag{16}$$

and

$$\frac{\partial^2 \log |\mathbf{F}|}{\partial x_i \partial x_j} = \text{tr}\left(\mathbf{F}^{-1}\frac{\partial^2 \mathbf{F}}{\partial x_i \partial x_j}\right) - \text{tr}\left(\mathbf{F}^{-1}\frac{\partial \mathbf{F}}{\partial x_i}\mathbf{F}^{-1}\frac{\partial \mathbf{F}}{\partial x_j}\right) \tag{17}$$

(see Harville, 1997), the expressions for the derivatives are as follows (all r.h.s. terms transformed versions):

$$\frac{\partial \log |\mathbf{S}|}{\partial \rho_j} = \lambda_j \text{tr}\left(\mathbf{S}^{-1}\mathbf{S}_j\right)$$

and

$$\frac{\partial^2 \log |\mathbf{S}|}{\partial \rho_i \partial \rho_j} = \delta_j^i \lambda_i \text{tr}\left(\mathbf{S}^{-1}\mathbf{S}_i\right) - \lambda_i \lambda_j \text{tr}\left(\mathbf{S}^{-1}\mathbf{S}_i\mathbf{S}^{-1}\mathbf{S}_j\right).$$

Appendix C: The derivatives of $\hat{\boldsymbol{\beta}}$ using implicit differentiation

When full Newton is used in place of Fisher scoring to obtain $\hat{\boldsymbol{\beta}}$, then there is no computational advantage in iterating for the derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. $\boldsymbol{\rho}$ (as in Wood, 2008), rather than exploiting the implicit function theorem to get them directly by implicit differentiation. This is because Newton based PIRLS requires exactly the same quantities as implicit differentiation. This appendix provides the details.

Define

$$D_p = D(\boldsymbol{\beta}) + \sum_m e^{\rho_m} \boldsymbol{\beta}^{\mathsf{T}} \mathbf{S}_m \boldsymbol{\beta},$$

and note that in this appendix some care must be taken to distinguish total derivatives of D_p, which encompass all variability with respect to a variable, as opposed to partial derivatives of the expression for D_p which ignore dependence of $\hat{\boldsymbol{\beta}}$ on $\boldsymbol{\rho}$.

The *partial* derivatives of D_p

$$\frac{\partial D}{\partial \beta_r} = -2\sum_i \omega_i \frac{(y - \mu_i)}{V(\mu_i) g'(\mu_i)} \mathbf{X}_{ir} \text{ and } \frac{\text{d}\mu_i}{\text{d}\beta_r} = \frac{X_{ir}}{g'(\mu_i)},$$

from which it follows (after some calculation) that

$$\frac{\partial^2 D}{\partial \beta_r \partial \beta_m} = \sum_i 2 w_i X_{im} X_{ir}$$

where w_i is the Newton version. Consequently

$$\frac{\partial^3 D}{\partial \beta_r \partial \beta_m \partial \beta_l} = \sum_i \frac{\mathrm{d} w_i}{\mathrm{d} \eta_i} X_{im} X_{ir} X_{il}.$$

Note that the *partials* of D w.r.t. $\boldsymbol{\rho}$ are zero.

Turning to $P = \sum_m e^{\rho_m} \boldsymbol{\beta}^\mathsf{T} \mathbf{S}_m \boldsymbol{\beta}$ (so $D_p = D + P$) we have

$$\nabla_\beta P = 2 \sum_m e^{\rho_m} \mathbf{S}_m \boldsymbol{\beta} \text{ and } \nabla^2_\beta P = 2 \sum_m e^{\rho_m} \mathbf{S}_m.$$

Furthermore

$$\frac{\partial \nabla_\beta P}{\partial \rho_j} = 2 e^{\rho_j} \mathbf{S}_j \boldsymbol{\beta} \quad \text{and} \quad \frac{\partial^2 \nabla_\beta P}{\partial \rho_j \partial \rho_k} = 2 \delta^k_j e^{\rho_j} \mathbf{S}_j \boldsymbol{\beta}, \text{ while } \frac{\partial \nabla^2_\beta P}{\partial \rho_j} = 2 e^{\rho_j} \mathbf{S}_j.$$

The derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t. $\boldsymbol{\rho}$

$\hat{\boldsymbol{\beta}}$ is the solution to

$$\frac{\mathrm{d} D_p}{\mathrm{d} \beta_r} = 0.$$

Since the above equation always holds at $\hat{\beta}$, we have

$$\frac{\mathrm{d}^2 D_p}{\mathrm{d} \beta_r \mathrm{d} \rho_j} = \sum_m \frac{\partial^2 D_p}{\partial \beta_r \partial \beta_m} \frac{\mathrm{d} \beta_m}{\mathrm{d} \rho_j} + \frac{\partial^2 D_p}{\partial \beta_r \partial \rho_j} = 0,$$

at $\hat{\boldsymbol{\beta}}$. i.e.

$$\frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_j} = - \left[\frac{\partial^2 D_p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\mathsf{T}} \right]^{-1} \frac{\partial \nabla_\beta D_p}{\partial \rho_j}.$$

Differentiating again we get

$$\frac{\mathrm{d}^3 D_p}{\mathrm{d} \beta_r \mathrm{d} \rho_j \mathrm{d} \rho_k} = \sum_l \sum_m \frac{\partial^3 D_p}{\partial \beta_r \partial \beta_m \partial \beta_l} \frac{\mathrm{d} \beta_m}{\mathrm{d} \rho_j} \frac{\mathrm{d} \beta_l}{\mathrm{d} \rho_k} + \sum_m \frac{\partial^3 D_p}{\partial \beta_r \partial \beta_m \partial \rho_k} \frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_j} + \sum_m \frac{\partial^2 D_p}{\partial \beta_r \partial \beta_m} \frac{\mathrm{d}^2 \beta_m}{\mathrm{d} \rho_j \mathrm{d} \rho_k}$$
$$+ \sum_m \frac{\partial^3 D_p}{\partial \beta_r \partial \beta_m \partial \rho_j} \frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_k} + \frac{\partial^3 D_p}{\partial \beta_r \partial \rho_j \partial \rho_k} = 0$$

Now

$$\frac{\mathrm{d} \boldsymbol{\eta}}{\mathrm{d} \rho_j} = \mathbf{X} \frac{\mathrm{d} \boldsymbol{\beta}}{\mathrm{d} \rho_j},$$

so using the expression for the third partial of D/D_p w.r.t. $\boldsymbol{\rho}$, and re-arranging we get

$$\begin{aligned} \frac{\mathrm{d}^2 \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_j \mathrm{d} \rho_k} &= - \left[\frac{\partial^2 D_p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\mathsf{T}} \right]^{-1} \left\{ \frac{\partial^2 \nabla_\beta D_p}{\partial \rho_j \partial \rho_k} + \mathbf{X}^\mathsf{T} \mathbf{f}^{jk} + 2 e^{\rho_j} \mathbf{S}_j \frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_k} + 2 e^{\rho_k} \mathbf{S}_k \frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_j} \right\} \\ &= \delta^k_j \frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_k} - \left[\frac{\partial^2 D_p}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\mathsf{T}} \right]^{-1} \left\{ \mathbf{X}^\mathsf{T} \mathbf{f}^{jk} + 2 e^{\rho_j} \mathbf{S}_j \frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_k} + 2 e^{\rho_k} \mathbf{S}_k \frac{\mathrm{d} \hat{\boldsymbol{\beta}}}{\mathrm{d} \rho_j} \right\} \end{aligned}$$

where

$$f^{jk}_i = \frac{\mathrm{d} \eta_i}{\mathrm{d} \rho_j} \frac{\mathrm{d} \eta_i}{\mathrm{d} \rho_k} \frac{\mathrm{d} w_i}{\mathrm{d} \eta_i}.$$

Note that the required inverse is $\mathbf{P}\mathbf{P}^\mathsf{T}/2$ (with derivatives of dropped parameters set to zero by this choice).

Appendix D: Derivatives of w

In this appendix primes denote differentiation w.r.t. μ_i. First the derivatives of α_i are useful

$$\alpha_i' = -\left(\frac{V_i'}{V_i} + \frac{g_i''}{g_i'}\right) + (y_i - \mu_i)\left(\frac{V_i''}{V_i} - \frac{V_i'^2}{V_i^2} + \frac{g_i'''}{g_i'} - \frac{g_i''^2}{g_i'^2}\right)$$

and

$$\alpha_i'' = -2\left(\frac{V_i''}{V_i} - \frac{V_i'^2}{V_i^2} + \frac{g_i'''}{g_i'} - \frac{g_i''^2}{g_i'^2}\right) + (y_i - \mu_i)\left(\frac{V_i'''}{V_i} - \frac{3V_i'V_i''}{V_i^2} + \frac{2V_i'^3}{V_i^3} + \frac{g_i''''}{g_i'} - \frac{3g_i'''g_i''}{g_i'^2} + \frac{2g_i''^3}{g_i'^3}\right).$$

The key derivatives of w_i are then

$$\frac{\mathrm{d}w_i}{\mathrm{d}\eta_i} = \frac{w_i}{g_i'}\left(\frac{\alpha_i'}{\alpha_i} - \frac{V_i'}{V_i} - 2\frac{g_i''}{g_i'}\right)$$

and

$$\frac{\mathrm{d}^2 w_i}{\mathrm{d}\eta_i^2} = \frac{1}{w_i}\left(\frac{\mathrm{d}w_i}{\mathrm{d}\eta_i}\right)^2 - \frac{\mathrm{d}w_i}{\mathrm{d}\eta_i}\frac{g_i''}{g_i'^2} + \frac{w_i}{g_i'^2}\left(\frac{\alpha_i''}{\alpha_i} - \frac{\alpha_i'^2}{\alpha_i^2} - \frac{V_i''}{V_i} + \frac{V_i'^2}{V_i^2} - 2\frac{g'''}{g_i'} + 2\frac{g_i''^2}{g_i'^2}\right).$$

The derivatives of $\boldsymbol{\eta}$ w.r.t. $\boldsymbol{\rho}$ are obtained from the derivatives of $\hat{\boldsymbol{\beta}}$ w.r.t $\boldsymbol{\rho}$, so the derivatives of w_i w.r.t. $\boldsymbol{\rho}$ follow easily. Note that setting $\alpha_i \equiv 1$, and its derivatives to zero, recovers Fisher scoring.

Appendix E: The ML determinant term and derivatives

ML requires computation of $\log|\bar{\mathbf{X}}^{\mathsf{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}|$ and its derivatives (see section 2.1). This requires further work. First note that explicit formation and decomposition of $\sqrt{\mathbf{W}}\mathbf{X}\mathbf{U}_1$ would be wasteful. All that is needed is the (pivoted) QR decomposition

$$\mathbf{R}\mathbf{U}_1 = \bar{\mathbf{Q}}\bar{\mathbf{R}}$$

where $\mathbf{R}$ is from section 3.3. $\mathbf{R}$ (and $\mathbf{Q}_1$) should not be truncated here, even if there is rank deficiency: instead $\bar{\mathbf{R}}$ and $\bar{\mathbf{Q}}$ should be. It is then easy to show that

$$\bar{\mathbf{X}}^{\mathsf{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}} = \bar{\mathbf{R}}^{\mathsf{T}}(\mathbf{I} - 2\bar{\mathbf{Q}}^{\mathsf{T}}\mathbf{Q}_1^{\mathsf{T}}\mathbf{I}^{-}\mathbf{Q}_1\bar{\mathbf{Q}})\bar{\mathbf{R}}.$$

Forming the SVD

$$\mathbf{I}^{-}\mathbf{Q}_1\bar{\mathbf{Q}} = \bar{\mathbf{U}}\bar{\mathbf{D}}\bar{\mathbf{V}}^{\mathsf{T}},$$

define

$$\bar{\mathbf{P}} = \begin{pmatrix} \bar{\mathbf{R}}^{-1}\bar{\mathbf{V}}(\mathbf{I} - 2\bar{\mathbf{D}}^2)^{-1/2} \\ \mathbf{0} \end{pmatrix} \quad \text{and} \quad \bar{\mathbf{K}} = \mathbf{Q}_1\bar{\mathbf{Q}}\bar{\mathbf{V}}(\mathbf{I} - 2\bar{\mathbf{D}}^2)^{-1/2}.$$

Then $|\bar{\mathbf{X}}^{\mathsf{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}| = |\bar{\mathbf{R}}|^2|\mathbf{I} - 2\bar{\mathbf{D}}^2|$ and the expressions for the derivatives of $\log|\bar{\mathbf{X}}^{\mathsf{T}}\mathbf{W}\bar{\mathbf{X}} + \bar{\mathbf{S}}|$ are as in section 3.5.1, but with $\bar{\mathbf{P}}$ and $\bar{\mathbf{K}}$ in place of $\mathbf{P}$ and $\mathbf{K}$ and the $\mathbf{S}_k$ replaced by $\bar{\mathbf{S}}_k = \mathbf{U}_1^{\mathsf{T}}\mathbf{S}_k\mathbf{U}_1$ (pivoted in the same way as the $\bar{\mathbf{R}}$).

Appendix F:The Pearson Statistic

The derivatives of the Pearson statistic with respect to the coefficients are required. Wood (2008) provided these in a form which only holds under Fisher scoring. Here is the general form.

$$P = \sum_i P_i \text{ where } P_i = \frac{\omega_i(y_i - \mu_i)^2}{V_i}.$$

So we need

$$\frac{\mathrm{d}P_i}{\mathrm{d}\beta_j} = \frac{\mathrm{d}P_i}{\mathrm{d}\eta_i}X_{ij} \text{ and } \frac{\mathrm{d}^2P_i}{\mathrm{d}\beta_j\mathrm{d}\beta_k} = \frac{\mathrm{d}^2P_i}{\mathrm{d}\eta_i^2}X_{ij}X_{ik}.$$

The requisite derivatives are

$$\frac{\mathrm{d}P_i}{\mathrm{d}\eta_i} = -\frac{1}{g_i'}\left\{\frac{2\omega_i(y_i-\mu_i)}{V_i} + P_i\frac{V_i'}{V_i}\right\}$$

and

$$\frac{\mathrm{d}^2P_i}{\mathrm{d}\eta_i^2} = \frac{g_i''}{g_i'^3}\left\{\frac{2\omega_i(y_i-\mu_i)}{V_i} + P_i\frac{V_i'}{V_i}\right\} + \frac{1}{g_i'^2}\left\{\frac{2\omega_i}{V_i} + \frac{2\omega_i(y_i-\mu_i)}{V_i}\frac{V_i'}{V_i} - g_i'\frac{\mathrm{d}P_i}{\mathrm{d}\eta_i}\frac{V_i'}{V_i} - P_i\left(\frac{V_i''}{V_i} - \frac{V_i'^2}{V_i^2}\right)\right\}.$$

Appendix G: Derivatives of the saturated log-likelihood

When the scale parameter is fixed and known, as in the binomial and Poisson cases, then l_s is irrelevant and its derivative w.r.t. ϕ is zero. Otherwise l_s and derivatives are needed. Here are three common examples.

Gaussian

$l_s = -\log(\phi)/2 - \log(2\pi)/2$, $l_s' = -1/(2\phi)$ and $l_s'' = 1/(2\phi^2)$.

Inverse Gaussian

$l_s = -\log(\phi)/2 - \log(2\pi y^3)/2$, $l_s' = -1/(2\phi)$ and $l_s'' = 1/(2\phi^2)$.

Gamma

$l_s = -\log\Gamma(1/\phi) - \log(\phi)/\phi - 1/\phi - \log(y)$. Writing $\log\Gamma$ to mean the log gamma function (to be differentiated as a whole): $l_s' = \log\Gamma'(1/\phi)/\phi^2 + \log(\phi)/\phi^2$ and $l_s'' = -\log\Gamma''(1/\phi)/\phi^4 - 2\log\Gamma'(1/\phi)/\phi^3 + \{1 - 2\log(\phi)\}/\phi^3$. The `lgamma`, `digamma` and `trigamma` functions in R evaluate $\log\Gamma$, $\log\Gamma'$ and $\log\Gamma''$ respectively.

Appendix H: Derivatives of $\mathrm{tr}\,(\mathbf{F})$

Prediction error criteria, such as GCV, involve the effective degrees of freedom of a model defined as $\mathrm{tr}\,(\mathbf{F})$ where

$$\mathbf{F} = (\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S})^{-1}\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}.$$

To optimize such criteria using the method developed here requires differentiation of $\mathrm{tr}\,(\mathbf{F})$ w.r.t. the log smoothing parameters. Define $\mathbf{G} = \mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{S}$. Note that $\mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\sqrt{\bar{\mathbf{W}}} = \mathbf{P}\mathbf{K}^\mathsf{T}$, $\sqrt{\bar{\mathbf{W}}}\mathbf{X}\mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\sqrt{\bar{\mathbf{W}}} = \mathbf{K}\mathbf{K}^\mathsf{T}$ and $\mathbf{G}^{-1} = \mathbf{P}\mathbf{P}^\mathsf{T}$. Also define $\mathbf{T}_j$ and $\mathbf{T}_{jk}$ as in section 3.5.1 (and **not** as in Wood, 2008), and diagonal matrix $\mathbf{I}^+$ where $I_{ii}^+ = -1$ if $w_i < 0$ and $I_{ii}^+ = 1$ otherwise. Now $\mathbf{F} = \mathbf{P}\mathbf{K}^\mathsf{T}\mathbf{I}^+\sqrt{\bar{\mathbf{W}}}\mathbf{X}$ and

$$\frac{\partial\mathbf{F}}{\partial\rho_j} = -\mathbf{G}^{-1}\left(\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_j}\mathbf{X} + e^{\rho_j}\mathbf{S}_j\right)\mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} + \mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_j}\mathbf{X},$$

so that

$$\frac{\partial\mathrm{tr}\,(\mathbf{F})}{\partial\rho_j} = -\mathrm{tr}\left(\mathbf{K}\mathbf{K}^\mathsf{T}\mathbf{T}_j\mathbf{K}\mathbf{K}^\mathsf{T}\mathbf{I}^+\right) - e^{\rho_j}\mathrm{tr}\left(\mathbf{K}\mathbf{P}^\mathsf{T}\mathbf{S}_j\mathbf{P}\mathbf{K}^\mathsf{T}\mathbf{I}^+\right) + \mathrm{tr}\left(\mathbf{K}\mathbf{K}^\mathsf{T}\mathbf{T}_j\right).$$

Second derivatives are more tedious

$$\begin{aligned}\frac{\partial^2\mathbf{F}}{\partial\rho_j\partial\rho_k} = &\left[\mathbf{G}^{-1}\left(\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_j}\mathbf{X} + e^{\rho_j}\mathbf{S}_j\right)\mathbf{G}^{-1}\left(\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_k}\mathbf{X} + e^{\rho_k}\mathbf{S}_k\right)\mathbf{G}^{-1}\right]^{\ddagger}\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X}\\ &- \mathbf{G}^{-1}\left(\mathbf{X}^\mathsf{T}\frac{\partial^2\mathbf{W}}{\partial\rho_j\partial\rho_k}\mathbf{X} + \delta_j^k e^{\rho_j}\mathbf{S}_j\right)\mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\mathbf{W}\mathbf{X} - \mathbf{G}^{-1}\left(\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_j}\mathbf{X} + e^{\rho_j}\mathbf{S}_j\right)\mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_k}\mathbf{X}\\ &- \mathbf{G}^{-1}\left(\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_k}\mathbf{X} + e^{\rho_k}\mathbf{S}_k\right)\mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\frac{\partial\mathbf{W}}{\partial\rho_j}\mathbf{X} + \mathbf{G}^{-1}\mathbf{X}^\mathsf{T}\frac{\partial^2\mathbf{W}}{\partial\rho_j\partial\rho_k}\mathbf{X},\end{aligned}$$

where $[\mathbf{A}]^{\ddagger} = \mathbf{A} + \mathbf{A}^{\mathsf{T}}$. It follows that

$$\begin{aligned}\frac{\partial^2 \mathrm{tr}\left(\mathbf{F}\right)}{\partial\rho_j\partial\rho_k} = {} & 2\mathrm{tr}\left(\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{T}_k\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{T}_j\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{I}^{+}\right) + 2e^{\rho_j}\mathrm{tr}\left(\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{T}_k\mathbf{K}\mathbf{P}^{\mathsf{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathsf{T}}\mathbf{I}^{+}\right) \\ & + 2e^{\rho_k}\mathrm{tr}\left(\mathbf{K}\mathbf{P}^{\mathsf{T}}\mathbf{S}_k\mathbf{P}\mathbf{K}^{\mathsf{T}}\mathbf{T}_j\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{I}^{+}\right) + 2e^{\rho_k+\rho_j}\mathrm{tr}\left(\mathbf{K}\mathbf{P}^{\mathsf{T}}\mathbf{S}_k\mathbf{P}\mathbf{P}^{\mathsf{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathsf{T}}\mathbf{I}^{+}\right) \\ & - \mathrm{tr}\left(\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{T}_{jk}\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{I}^{+}\right) - \delta_j^k e^{\rho_j}\mathrm{tr}\left(\mathbf{K}\mathbf{P}^{\mathsf{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathsf{T}}\mathbf{I}^{+}\right) - 2\mathrm{tr}\left(\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{T}_k\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{T}_j\right) \\ & - e^{\rho_j}\mathrm{tr}\left(\mathbf{K}\mathbf{P}^{\mathsf{T}}\mathbf{S}_j\mathbf{P}\mathbf{K}^{\mathsf{T}}\mathbf{T}_k\right) - e^{\rho_k}\mathrm{tr}\left(\mathbf{K}\mathbf{P}^{\mathsf{T}}\mathbf{S}_k\mathbf{P}\mathbf{K}^{\mathsf{T}}\mathbf{T}_j\right) + \mathrm{tr}\left(\mathbf{K}\mathbf{K}^{\mathsf{T}}\mathbf{T}_{jk}\right).\end{aligned}$$

Although $\mathbf{K}$, $\mathbf{P}$ and the $\mathbf{T}$ matrices are all different to those in Wood (2008), and the $\mathbf{I}^{+}$ matrices did not feature there at all, it is still possible to use the tricks listed in Appendix C of Wood (2008) to evaluate these terms efficiently, with only minor adjustment.

Note that there is a strong argument for employing Fisher scoring based weights in place of Newton based weights in the definition of $\mathbf{F}$. This requires redefining $\mathbf{W}$, $\mathbf{T}_k$ and $\mathbf{T}_{jk}$ and setting $\mathbf{I}^{+}$ to $\mathbf{I}$, but otherwise the computations are identical. This change removes the possibility of $\mathbf{X}^{\mathsf{T}}\mathbf{W}\mathbf{X}$ having negative eigenvalues, which can occasionally lead to non-sensical computed effective degrees of freedom.

References

Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Donngarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney & D. Sorenson (1999) *LAPACK Users' Guide* (3rd ed.) SIAM, Philadelphia.

Anderssen, R.S. & P. Bloomfield (1974) A Time Series Approach to Numerical Differentiation *Technometrics*, 16(1), 69-75.

Breslow, N.E. & D.G. Clayton (1993) Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* 88, 9-25.

Brezger, A. & S. Lang (2006) Generalized structured additive regression based on Bayesian P-splines. *Computational Statistics and Data Analysis* 50, 967-991.

Brezger, A., T. Kneib & S. Lang (April 2007) *BayesX* 1.5.0 `http://www.stat.uni-muenchen.de/~bayesx`

Cline, A.K, C.B. Moler, G.W. Stewart & J.H. Wilkinson (1979) An Estimate for the Condition Number of a Matrix. *SIAM Journal of Numerical Analysis* 13, 293-309.

Craven P. & G. Wahba (1979) Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized cross validation. *Numerische Mathematik* 31, 377-403.

Davison, A.C. (2003) *Statistical Models*. Cambridge.

Demidenko, E. (2004) *Mixed Models: theory and applications*. Wiley.

Dunn, P.K. and G.K. Smith (2005) Series evaluation of Tweedie exponential dispersion model densities. *Statistics and Computing* 15, 267-280.

Efron, B. and D.V. Hinkley (1978) Assessing the accuracy of the maximum likelihood estimator: observed versus expected Fisher information. *Biometrika* 65, 457-487.

Eilers, P.H.C. and B.D. Marx (1996) Flexible Smoothing with B-splines and Penalties. *Statistical Science*, 11(2), 89-121.

Eilers, P.H.C. & B.D. Marx (2002) Generalized linear additive smooth structures. *Journal of computational and graphical statistics* 11(4), 758-783.

Escabias, M., A.M. Aguilera, and M.J. Valderrama (2004) Principal component estimation of functional logistic regression: discussion of two different approaches. *Nonparametric Statistics* 16, 365-384.

Fahrmeir, L., T. Kneib & S. Lang (2004) Penalized structured additive regression for space time data: A Bayesian perspective. *Statistica Sinica* 14, 731-761.

Fahrmeir, L. & S. Lang (2001) Bayesian inference for generalized additive mixed models based on Markov random field priors. *Applied Statistics* 50, 201-220.

Golub, G.H. & C.F. van Loan (1996) *Matrix Computations*. (3rd edition). Johns Hopkins University Press, Baltimore.

Green, P.J. & B.W. Silverman (1994) *Nonparametric Regression and Generalized Linear Models*. Chapman & Hall, London.

Gu, C. (1992) Cross validating non-Gaussian data. *Journal of Computational and Graphical Statistics* 1, 169-179.

Gu, C. (2002) Smoothing Spline ANOVA Models. Springer, New York.

Gu. C. & Y-J Kim (2002) Penalized likelihood regression: general formulation and efficient approximation. *The Canadian Journal of Statistics* 30(4), 619- 628.

Hall, P & J.D. Opsomer (2005) Theory for penalised spline regression *Biometrika* (2005) 92(1), 105-118

Härdle, W., P. Hall & J. S. Marron (1988) How Far Are Automatically Chosen Regression Smoothing Parameters From Their Optimum? *Journal of the American Statistical Association*, 83, 86-95.

Harville, D.A., (1977) Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistal Association* 72, 320-338.

Harville, D.A. (1997) *Matrix Algebra From a Statisticians Perspective*. Springer.

Hastie, T. & R. Tibshirani (1986) Generalized additive models (with discussion). *Statistical Science* 1, 297-318.

Hastie, T. & R. Tibshirani (1990) *Generalized additive models*. Chapman & Hall, London.

Hastie, T. & R. Tibshirani (1993) Varying-coefficient models. *Journal of the Royal Statistical Society, Series B* 55, 757-796.

Hurvich, C.M., J.S. Siminoff and C-L Tsai (1998) Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society, Series B* 60, 271-293.

Kalivas, J.H. (1997) Two Data Sets of Near Infrared Spectra. *Chemometrics and Intelligent Laboratory Systems*, 37, 255-259.

Kauermann, G. (2005) A note on smoothing parameter selection for penalized spline smoothing. *Journal of Statistical planning and inference* 127, 53-69.

Kauermann, G., T. Krivobokova and L. Fahrmeir (2009) Some asymptotic results on generalized penalized spline smoothing. *Journal of the Royal Statistical Society, Series B* 71, 487-503.

Kimeldorf, G and G. Wahba (1970) A correspondence between Bayesian estimation of stochastic processes and smoothing by splines. *Annals of Mathematical Statistics* 41: 495-502.

Kohn, R., C. F. Ansley and D. Tharm (1991) The Performance of Cross-Validation and Maximum Likelihood Estimators of Spline Smoothing Parameters. *Journal of the American Statistical Association*, 86, 1042-1050.

Krivobokova, T., C.M. Crainiceanu & G. Kauermann (2008) Fast Adaptive Penalized Splines. *Journal of Computational and Graphical Statistics*. 17(1) 1-20.

Laird N.M. & J.H. Ware (1982) Random-Effects Models for Longitudinal Data. *Biometrics* 38(4), 963-974.

Lang, S & A. Brezger (2004) Bayesian P-splines. *Journal of Computational and Graphical Statistics* 13, 183-212.

Marx B. D. & P.H. Eilers (1998) Direct generalized additive modeling with penalized likelihood. *Computational Statistics and Data Analysis* 28, 193-209.

Marx B. D. & P.H. Eilers (1999) Generalized Linear Regression on Sampled Signals and Curves: A P-Spline Approach *Technometrics* 41(1), 1-13.

Monahan J.F. (2001) *Numerical Methods of Statistics*. Cambridge.

Nelder, J.A. & R.W.M. Wedderburn (1972) Generalized linear models. *Journal of the Royal Statistical Society, Series A* 135, 370-384

Nocedal, J. & S.J. Wright (2006) *Numerical Optimization* (2nd edition). Springer.

Parker, R. and J. Rice (1985) Discussion of Silverman (1985) *Journal of the Royal Statistical Society, Series B* 47(1), 41-42.

Patterson, H.D. and R. Thompson (1971) Recovery of interblock information when block sizes are unequal. *Biometrika* 58, 545-554.

R Core Development Team (2008) *R 2.8.1: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna.

Ramsay, J.O. & B.W. Silverman (2005) *Functional Data Analysis.* Springer.

Reiss, P.T. & R.T. Ogden (2009) Smoothing parameter selection for a class of semiparametric linear models. *Journal of the Royal Statistical Society, Series B* 71, 505-524.

Reiss, P.T. & R.T. Ogden (2007) Functional principal component regression and functional partial least squares. *Journal of the American Statistical Association* 102, 984-996.

Rue, H., S. Martino and N. Chopin (2009) Approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society, Series B* 71, 319-392.

Ruppert, D., M.P. Wand & R.J. Carroll (2003) *Semiparametric Regression.* Cambridge.

Silverman, B.W. (1985) Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *Journal of the Royal Statistical Society, Series B* 47, 1-53.

Tutz, G. & H. Binder (2006) Generalized Additive Modeling with Implicit Variable Selection by Likelihood-Based Boosting *Biometrics* 62, 961-971.

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. in *Statistics: Applications and New Directions, Proceedings of the Indian Statistical Institute Golden Jubilee International Conference* (Eds. J. K. Ghosh and J. Roy), pp. 579-604. Calcutta: Indian Statistical Institute.

Venables. W.N. and B.D Ripley (2002) *Modern Applied Statistics with S* (4th ed) Springer.

Wahba, G. & S. Wold (1975) A completely automatic French curve: Fitting spline functions by cross-validation. *Communications in Statistics* 4, 125-141.

Wahba, G. (1980) Spline bases, regularization and generalized cross validation for solving approximation problems with large quantities of noisy data. in E. Cheney (ed) *Approximation Theory III* Academic Press, London.

Wahba, G. (1983) Bayesian confidence intervals for the cross validated smoothing spline. *Journal of the Royal Statistical Society, Series B* 45, 133-150.

Wahba, G. (1985) A Comparison of GCV and GML for Choosing the Smoothing Parameter in the Generalized Spline Smoothing Problem. *The Annals of Statistics*, Vol. 13(4), 1378-1402.

Wahba, G (1990) *Spline models for observational data.* SIAM, Philadelphia.

Watkins, D.S. (1991) *Fundamentals of Matrix Computations.* Wiley, New York.

Wehrens R. & B-H Mevik (2007). pls: Partial Least Squares Regression (PLSR) and Principal Component Regression (PCR). R package version 2.1-0. `http://mevik.net/work/software/pls.html`

Wood, S.N. (2003) Thin plate regression splines. *Journal of the Royal Statistical Society, Series B* 65, 95-114.

Wood, S.N. (2004) Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association* 99, 673-686.

Wood, S.N. (2006) *Generalized Additive Models: An Introduction with R* CRC/Chapman & Hall.

Wood, S.N. (2008) Fast stable direct fitting and smoothness selection for generalized additive models. *Journal of the Royal Statistical Society, Series B* 70, 495-518.