Estimating Predictive Variances with Kernel Ridge Regression

Gavin C. Cawley*, Nicola L. C. Talbot* and Olivier Chapelle†

*School of Computing Sciences
University of East Anglia
Norwich NR4 7TJ, U. K.
{gcc,nlct}@cmp.uea.ac.uk

†Max Plank Institute
for Biological Cybernetics
72076 Tübingen, Germany
olivier.chapelle@teubingen.mpg.de

Abstract. In many regression tasks, in addition to an accurate estimate of the conditional mean of the target distribution, an indication of the predictive uncertainty is also required. There are two principal sources of this uncertainty: the noise process contaminating the data and the uncertainty in estimating the model parameters based on a limited sample of training data. Both of them can be summarised in the *predictive variance* which can then be used to give confidence intervals. In this paper, we present various schemes for providing predictive variances for kernel ridge regression, especially in the case of a heteroscedastic regression, where the variance of the noise process contaminating the data is a smooth function of the explanatory variables. The use of leave-one-out cross-validation is shown to eliminate the bias inherent in estimates of the predictive variance. Results obtained on all three regression tasks comprising the predictive uncertainty challenge demonstrate the value of this approach.

1 Introduction

The standard framework for regression is the following: Given a training set

$$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{\ell}, \quad \boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d, \quad y_i \in \mathbb{R}, \tag{1}$$

the goal is to infer a function $\mu(\boldsymbol{x})$ which predicts the best value (in the least squares sense) of the target function on any test point $\boldsymbol{x}$. However, in some situations, it is also useful to know the confidence of this prediction. For this reason, we also want to infer a function $\sigma(\boldsymbol{x})$ corresponding to the uncertainty of our prediction. For instance, the result of our prediction could be a statement of the form: "with 95% confidence, I think that the target associated with point $\boldsymbol{x}$ is in the interval $[\mu(\boldsymbol{x}) - 2\sigma(\boldsymbol{x}), \mu(\boldsymbol{x}) + 2\sigma(\boldsymbol{x})]$". It is important to note that this uncertainty comes from two independent components:

1. The noise in the data
2. The uncertainty in the estimation of the target function

Typically, the first contribution is preponderant when there are a lot of training data, while the second one becomes more important when they are few training data. Let us illustrate this by two extreme examples. First, imagine that $\mathcal{X} = \boldsymbol{x}_0$ and $P(y|\boldsymbol{x}_0)$ is normally distributed with mean 0 and variance σ. After seeing ℓ examples, the empirical mean is near from the true target (0 in this case and the distance is of the order $\frac{\sigma}{\sqrt{\ell}}$). Thus, when ℓ is large, we can predict the target value very accurately (i.e. the conditional mean), but because of the noise, we are not so sure about the target associated with an unseen test point. Another extreme example is the following: suppose that we know that there is no noise in the data, but that we are given a test point which is infinitely far away from the other training samples. Then, we are just completely unsure of the conditional mean. In summary, one can say that the uncertainty of the prediction is the sum of two terms:

$$\text{Uncertainty in the conditional mean } + \text{ estimated noise level.}$$

In this paper, we will try to estimate this uncertainty directly, by considering that the goal is to infer a function from $\mathcal{X}$ to $\mathbb{R} \times \mathbb{R}^+$, $\boldsymbol{x} \mapsto (\mu(\boldsymbol{x}),\ \sigma(\boldsymbol{x}))$ where the loss associated to a test point $(\boldsymbol{x}, y)$ is

$$\log \sigma^2(\boldsymbol{x}) + \frac{[\mu(\boldsymbol{x}) - y]^2}{\sigma^2(\boldsymbol{x})}. \tag{2}$$

Let us now be more precise by giving the definitions of the following quantities:

Conditional mean This is true mean $E_{y|\boldsymbol{x}} y$ where the expectation is taken with respect to the true data generating process.

Predictive mean We define this quantity as $\mu(\boldsymbol{x})$, the first output of the function being inferred. This is an estimate of the conditional mean given a training set.

Conditional variance We do not define it as the true noise level but in association with a predictive mean as

$$E_{y|\boldsymbol{x}}(y - \mu(\boldsymbol{x}))^2 = (E_{y|\boldsymbol{x}} y - \mu(\boldsymbol{x}))^2 + E_{y|\boldsymbol{x}}(y - E_{y|\boldsymbol{x}} y)^2.$$

Predictive variance Similarly to the predictive mean, this is defined as $\sigma^2(\boldsymbol{x})$, the square of the second output argument of the inferred function.

At this point, we would like to make the following remarks:

- The reason for considering the loss (2) is that it is (up to a constant) the negative log probability of a test point under the Gaussian predictive distribution with mean $\mu(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x})$, as used in [1].
- From the above definitions, the best predictive mean and variance for the loss (2) are the conditional mean and variance.

– The loss might seem arbitrary and from a decision theory point of view, one should consider the loss associated with the action taken based on the prediction $(\mu(\boldsymbol{x}), \sigma(\boldsymbol{x}))$. However, this still seems a reasonable "generic" loss. More generally, it is worth noting that a loss function is always necessary whenever a prediction is required.

– Instead of computing a predictive mean and variance, one could compute a predictive distribution (and the loss would be negative log predictive probability). But estimating a function instead of two real numbers is a more complicated inference problem and for the sake of simplicity, we do not consider it here. Note that in binary classification, the problem is much simpler as there is only real number to estimate, $P(y = 1|\boldsymbol{x})$.

The algorithm we propose in this paper alternates between updates of the predictive mean and of the predictive variance. For fixed σ, the predictive mean μ is modelled using an heteroscedastic kernel ridge regression algorithm. Heteroscedastic here just means that the error on a given training point $\boldsymbol{x}_i$ is weighted by $\sigma^{-2}(\boldsymbol{x}_i)$ (cf equation (2)). For fixed μ, a regression is performed on $\log \sigma$ in order to minimise (2). Again a regularised kernel regression algorithm is used for this purpose. Note that for learning the conditional variance, one should not use the same set of training points as the one used to learn the conditional mean since $(\mu(\boldsymbol{x}_i) - y_i)^2$ is not an unbiased estimator of the conditional variance at $\boldsymbol{x}_i$ if this point has been used to learn μ. Instead of a considering a "fresh" training set, we will use the leave-one-out procedure.

The basic algorithm that we use is Kernel Ridge Regression [2]. Considering the strong link of this algorithm with Gaussian Processes [3], the reader might wonder why we do not use this latter to estimate the predictive variances. The two reasons for this are:

1. We consider the more general case of heteroscedastic noise (i.e. whose variance depends on the input location).
2. We aim at showing that predictive variances can be calculated in a *non Bayesian* framework. However, we do not pretend that this approach is superior to the Bayesian approach. One of our main motivation is to answer the usual Bayesian criticism that standard non Bayesian methods do not provide predictive variances.

2 Kernel Ridge Regression

Kernel ridge regression (KRR) [2], or equivalently the least-squares support vector machine (LS-SVM) [4] provides perhaps the most basic form of kernel learning method. Given labelled training data (1), the kernel ridge regression method constructs a linear model $\mu(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}) + b$ in a fixed feature space $\mathcal{F}$ given by a fixed transformation of the input space, $\boldsymbol{\phi}(\boldsymbol{x}) : \mathcal{X} \to \mathcal{F}$. However, rather than specifying $\mathcal{F}$ directly, it is induced by a positive-definite kernel function [5]

$\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, which defines the inner product between vectors evaluated in $\mathcal{F}$, i.e. $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}')$. Kernel functions typically used in kernel learning methods include the spherical or isotropic Gaussian radial basis function (RBF) kernel,

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\kappa \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right\} \tag{3}$$

where κ is a kernel parameter, controlling the locality of the kernel, and the anisotropic Gaussian RBF kernel,

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\sum_{i=1}^{d} \kappa_i \left[x_i - x_i'\right]^2\right\} \tag{4}$$

which includes separate scale parameters, $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \ldots, \kappa_d)$, for each input dimension. The "kernel trick" allows us to create powerful linear models in high, or even infinite-dimensional feature spaces, using only finite dimensional quantities, such as the kernel or Gram matrix, $\boldsymbol{K} = [k_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^{\ell}$ (for a more detailed introduction to kernel learning methods, see [6, 7]). The kernel ridge regression method assumes that the data represent realisations of the output of some deterministic process that have been corrupted by additive Gaussian noise with zero mean and fixed variance, i.e.

$$y_i = \mu(\boldsymbol{x}_i) + \varepsilon_i, \qquad \text{where} \qquad \varepsilon_i \sim \mathcal{N}(0,\ \sigma^2), \qquad \forall\ i \in \{1, 2, \ldots, \ell\}.$$

As in conventional linear ridge regression [8], the optimal model parameters $(\boldsymbol{w}, b)$ are determined by minimising a regularised loss function representing the penalised negative log-likelihood of the data,

$$\frac{1}{2}\gamma ||\boldsymbol{w}||^2 + \sum_{i=1}^{\ell} \left[\mu(\boldsymbol{x}_i) - y_i\right]^2.$$

The parameter γ can either be interpreted as a regularisation parameter or as an inverse noise variance. As shown in a more general setting in Section 3, the optimal $\boldsymbol{w}$ can be expressed as $\boldsymbol{w} = \sum_{i=1}^{\ell} \alpha_i \phi(\boldsymbol{x}_i)$, where $\boldsymbol{\alpha}$ is found by solving the following linear system,

$$\begin{bmatrix} \boldsymbol{K} + \gamma \boldsymbol{I} & \boldsymbol{1} \\ \boldsymbol{1}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}.$$

2.1 A Simple Model for Heteroscedastic Data

The kernel ridge regression model assumes the target data represent realisations of a deterministic system that have been corrupted by a Gaussian noise process with zero mean and constant (homoscedastic) variance. This is unrealistic in some practical applications, where the variance of the noise process is likely to be dependent in some way on the explanatory variables. For example, in environmental applications, the variability in the intensity of sunlight reaching

ground level is more variable in Spring, Summer and Autumn as, at least in the United Kingdom, the Winter sky is predominantly overcast. A less restrictive approach is based on the assumption of a heteroscedastic, where the variance of the Gaussian noise is made a function of the explanatory variables. It is well known that for a model trained to minimise the squared error, the output approximates the conditional mean of the target data. Therefore, if we then train a second kernel ridge regression model to predict the squared residuals of the first, the output of the second model will be an estimate of the conditional mean of the squared residuals, i.e. the conditional variance of the target distribution. This method was suggested in the case of multi-layer perceptron networks (see e.g. [9]) by Satchwell [10] and applied to the problem of automotive engine calibration by Lowe and Zapart [11].

There are two problems with this method: the first one is that the squared residual is not an estimate of the conditional variance. Indeed, imagine that some over-fitting occurred while modelling the conditional mean: the squared residuals can then be very small not reflecting the true conditional variance. The second problem is that while modelling the conditional mean, the amount of regularisation is the same over all the space, while intuitively, one would like to regularise more in noisy regions. The first problem will be addressed in section 5 and the second one in the following section.

3 Heteroscedastic Kernel Ridge Regression

A more natural method of modelling heteroscedastic data fits the models of the predictive mean and predictive variance, or equivalently the predictive standard deviation, simultaneously, using a likelihood function corresponding to a Gaussian noise process with data-dependant variance, i.e.

$$p(\mathcal{D}|\boldsymbol{w}) = \prod_{i=1}^{\ell} \frac{1}{\sqrt{2\pi}\sigma(\boldsymbol{x}_i)} \exp\left\{ -\frac{[\mu(\boldsymbol{x}_i) - y_i]^2}{2\sigma^2(\boldsymbol{x}_i)} \right\}$$

where $\boldsymbol{w}$ represents the parameters of the combined model. A linear model of the conditional mean, $\mu(\boldsymbol{x}) = \boldsymbol{w}^\mu \cdot \boldsymbol{\phi}^\mu(\boldsymbol{x}) + b^\mu$ is then constructed in a feature space $\mathcal{F}^\mu$ corresponding to a positive definite kernel $\mathcal{K}^\mu(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}^\mu(\boldsymbol{x}) \cdot \boldsymbol{\phi}^\mu(\boldsymbol{x}')$. Similarly, the standard deviation being a strictly positive quantity, a linear model of the logarithm of the predictive standard deviation, $\log \sigma(\boldsymbol{x}) = \boldsymbol{w}^\sigma \cdot \boldsymbol{\phi}^\sigma(\boldsymbol{x}) + b^\sigma$ is constructed in a second feature space, $\mathcal{F}^\sigma$, induced by a second positive-definite kernel $\mathcal{K}^\sigma(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}^\sigma(\boldsymbol{x}) \cdot \boldsymbol{\phi}^\sigma(\boldsymbol{x}')$. The optimal model parameters, $(\boldsymbol{w}^\mu, b^\mu . \boldsymbol{w}^\sigma, b^\sigma)$, are determined by minimising a penalised negative log-likelihood objective function,

$$L = \frac{1}{2}\gamma^\mu \|\boldsymbol{w}^\mu\|^2 + \frac{1}{2}\gamma^\sigma \|\boldsymbol{w}^\sigma\|^2 + \frac{1}{2}\sum_{i=1}^{\ell} \left\{ \log \sigma(\boldsymbol{x}_i) + \frac{[\mu(\boldsymbol{x}_i) - y_i]^2}{2\sigma^2(\boldsymbol{x}_i)} \right\}, \qquad (5)$$

with regularisation parameters, γ^μ and γ^σ, providing individual control over the complexities of the models of the predictive mean and standard deviation respectively (c.f. [12, 13]). Note that (5) is the regularised objective function associated

with the loss (2). The use of a heteroscedastic loss leads to an important interaction between the data misfit and regularisation terms comprising the objective function : The squared error term is now weighted according to the estimated local variance of the data. As a result, the influence of the regularisation term is now increased in areas of high predictive variance. This is in agreement with our intuition that more flexible models are more easily justified where amplitude of the noise contaminating the data is low and meaningful variations in the underlying deterministic system we hope to model are obscured to a lesser degree. The $\log \sigma(\boldsymbol{x}_i)$ term penalises unduly high predictive standard deviations. It should be noted that it is possible for the negative log-likelihood term in (5) to go to minus infinity if the predictive variance goes to zero and $\mu(\boldsymbol{x}_i) = y_i$. One could circumvent this problem by adopting a suitable prior on b^σ, to indicate that we do not believe in very small predictive variances. However, this might not be enough and a more principled solution is presented in section 5. From a theoretical point of view, it is known that the ERM principle is consistent [14], so it might seem surprising that the minimiser of (5) would not yield functions giving a good test error (2), as the number of points goes to infinity. The reason why ERM could fail here is that the *loss is unbounded* and thus the convergence results about ERM do not apply.

A straight-forward extension of the representer theorem [15–17] indicates that the minimiser of this objective function can be expressed in the form of a pair of kernel expansions: For the model of the predictive mean,

$$\boldsymbol{w}^\mu = \sum_{i=1}^{\ell} \alpha_i^\mu \boldsymbol{\phi}^\mu(\boldsymbol{x}_i) \qquad \Longrightarrow \qquad \mu(\boldsymbol{x}) = \sum_{i=1}^{\ell} \alpha_i^\mu \mathcal{K}^\mu(\boldsymbol{x}_i, \boldsymbol{x}) + b^\mu,$$

and similarly for the model of the predictive standard deviation,

$$\boldsymbol{w}^\sigma = \sum_{i=1}^{\ell} \alpha_i^\sigma \boldsymbol{\phi}^\sigma(\boldsymbol{x}_i) \qquad \Longrightarrow \qquad \log \sigma(\boldsymbol{x}) = \sum_{i=1}^{\ell} \alpha_i^\sigma \mathcal{K}^\sigma(\boldsymbol{x}_i, \boldsymbol{x}) + b^\sigma.$$

The resulting model is termed a heteroscedastic kernel ridge regression (HKRR) machine [18, 17] (see also [19]). An efficient iterative training algorithm for this model alternates between updates of the model of the predictive mean and updates of the model of the predictive standard deviation.

3.1 Updating the Predictive Mean

Ignoring any terms in the objective function (5) that do not involve $\boldsymbol{w}^\mu$ or b^μ, a simplified cost function is obtained, which is used to update the parameters of the model of the predictive mean, $\mu(\boldsymbol{x}_i)$,

$$L^\mu = \frac{1}{2}\gamma^\mu \|\boldsymbol{w}^\mu\|^2 + \frac{1}{2}\sum_{i=1}^{\ell} \lambda_i \left[\mu(\boldsymbol{x}_i) - y_i\right]^2 \tag{6}$$

where $\lambda_i^{-1} = 2\sigma^2(\boldsymbol{x}_i)$. This is essentially equivalent to the cost function for a weighted least-squares support vector machine (LS-SVM) [4]. Minimising (6) can be recast in the form of a constrained optimisation problem,

$$\min \mathcal{J} = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\gamma^\mu}\sum_{i=1}^{\ell}\lambda_i\varepsilon_i^2 \tag{7}$$

subject to

$$y_i = \boldsymbol{w}^\mu \cdot \boldsymbol{\phi}^\mu(\boldsymbol{x}_i) + b^\mu + \varepsilon_i, \qquad \forall\ i \in \{1, 2, \ldots, \ell\}, \tag{8}$$

The Lagrangian for this optimisation problem gives the unconstrained minimisation problem,

$$\mathcal{L} = \frac{1}{2}\|\boldsymbol{w}^\mu\|^2 + \frac{1}{2\gamma^\mu}\sum_{i=1}^{\ell}\lambda_i\varepsilon_i^2 - \sum_{i=1}^{\ell}\alpha_k^\mu\left\{\boldsymbol{w}^\mu \cdot \boldsymbol{\phi}^\mu(\boldsymbol{x}_i) + b^\mu + \varepsilon_i - y_i\right\}, \tag{9}$$

where $\boldsymbol{\alpha}^\mu = (\alpha_1^\mu, \alpha_2^\mu, \ldots, \alpha_\ell^\mu) \in \mathbb{R}^\ell$ is a vector of Lagrange multipliers.

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}^\mu} = \boldsymbol{0} \implies \boldsymbol{w}^\mu = \sum_{i=1}^{\ell}\alpha_i^\mu\boldsymbol{\phi}^\mu(\boldsymbol{x}_i) \tag{10}$$

$$\frac{\partial\mathcal{L}}{\partial b^\mu} = 0 \implies \sum_{i=1}^{\ell}\alpha_i^\mu = 0 \tag{11}$$

$$\frac{\partial\mathcal{L}}{\partial\varepsilon_i} = 0 \implies \alpha_i^\mu = \frac{\lambda_i\varepsilon_i}{\gamma^\mu}, \quad \forall_i \in \{1, 2, \ldots, \ell\} \tag{12}$$

Using (10) and (12) to eliminate $\boldsymbol{w}$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_\ell)$, from (9), we find that

$$\sum_{j=1}^{\ell}\alpha_j^\mu\boldsymbol{\phi}^\mu(\boldsymbol{x}_j)\cdot\boldsymbol{\phi}^\mu(\boldsymbol{x}_i) + b^\mu + \frac{\gamma^\mu\alpha_i^\mu}{\lambda_i} = y_i \qquad \forall\ i \in \{1, 2, \ldots, \ell\} \tag{13}$$

Noting that $\mathcal{K}^\mu(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}^\mu(\boldsymbol{x})\cdot\boldsymbol{\phi}^\mu(\boldsymbol{x}')$, the system of linear equations can be written more concisely in matrix form as

$$\begin{bmatrix} \boldsymbol{K}^\mu + \gamma^\mu\boldsymbol{Z} & \boldsymbol{1} \\ \boldsymbol{1}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha}^\mu \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix},$$

where $\boldsymbol{K}^\mu = \left[k_{ij}^\mu = \mathcal{K}^\mu(\boldsymbol{x}_i, \boldsymbol{x}_j)\right]_{i,j=1}^{\ell}$ and $\boldsymbol{Z} = \operatorname{diag}\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_\ell^{-1}\}$. The parameters for the model of the predictive mean can then be obtained with a computational complexity of $\mathcal{O}(\ell^3)$ operations.

3.2 Updating the Predictive Standard Deviation

Similarly, neglecting terms in the objective function (5) that do not involve $\boldsymbol{w}^\sigma$ or b^σ, a simplified cost function is obtained, which is used to update the parameters

of the model of the predictive standard deviation, $\sigma(\boldsymbol{x}_i)$, dividing through by γ^σ,

$$L^\sigma = \frac{1}{2}\|\boldsymbol{w}^\sigma\|^2 + \frac{1}{2\gamma^\sigma}\sum_{i=1}^{\ell}\left[z_i + \xi_i \exp\{-2z_i\}\right], \tag{14}$$

where $\xi_i = \frac{1}{2}\left[\mu(\boldsymbol{x}_i) - y_i\right]^2$ and $z_i = \log\sigma(\boldsymbol{x}_i)$. The reason for this latter re-parametrisation is that (14) yields an *unbounded* and *convex* optimisation problem.

A closed form expression for the minimum of this objective function is not apparent, and so it is minimised via an iteratively re-weighted least-squares (IRWLS) procedure [20], which is effectively equivalent to a Newton's method. Indeed, at each iteration, a quadratic approximation of the objective function around the solution is performed and this quadratic approximation is minimised analytically, yielding an updated solution. Consider the negative log-likelihood for a single training pattern,

$$l_i = z_i + \xi_i \exp\{-2z_i\},$$

with first and second derivatives, with respect to z_i, given by

$$\frac{\partial l_i}{\partial z_i} = 1 - 2\xi_i \exp\{-2z_i\} \qquad \text{and} \qquad \frac{\partial^2 l_i}{\partial z_i^2} = 4\xi_i \exp\{-2z_i\}.$$

As we are interested only in minimising the negative log-likelihood, we substitute a weighted least-squares criterion, providing a local approximation of l_i only up to some arbitrary constant, C, i.e.

$$q_i = \beta_i[\eta_i - z_i]^2 \approx l_i + C,$$

Clearly, we require the gradient and curvature of q_i and l_i, with respect to z_i, to be identical, and therefore

$$\begin{aligned}
\frac{\partial^2 q_i}{\partial z_i^2} = \frac{\partial^2 l_i}{\partial z_i^2} &\implies \beta_i = 2\xi_i \exp\{-2z_i\}, \\
\frac{\partial q_i}{\partial z_i} = \frac{\partial l_i}{\partial z_i} &\implies \eta_i = z_i - \frac{1}{2\beta_i} + \frac{1}{2}.
\end{aligned}$$

The original objective function (14) for the model of the predictive standard deviation, can then be solved iteratively by alternating between updates of $\boldsymbol{\alpha}^\sigma$ and b^σ via a regularised weighted least-squares loss function,

$$\widetilde{L}^\sigma = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\gamma^\sigma}\sum_{i=1}^{\ell}\beta_i[\eta_i - z_i]^2, \tag{15}$$

and updates of the weighting coefficients, $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_\ell)$, and targets, $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_\ell)$, according to,

$$\beta_i = 4\xi_i \exp\{-2z_i\} \qquad \text{and} \qquad \eta_i = z_i - \frac{2}{\beta_i} + \frac{1}{2}. \tag{16}$$

The weighted least-squares problem (15) can also be solved via a system of linear equations, with a computational complexity of $\mathcal{O}(\ell^3)$ operations, using the methods described in section 3.1.

4 Model Selection

While efficient optimisation algorithms exist for the optimisation problems defining the primary model parameters for kernel machines, generalisation performance is also dependent on the values of a small set of hyper-parameters, in this case the regularisation and kernel parameters. The search for "good" values of these hyper-parameters is an activity known as model selection. A common model selection strategy seeks to minimise a cross-validation [21] estimate of some appropriate performance statistic, such as the mean squared error or negative log-likelihood. The k-fold cross-validation procedure partitions the available data into k disjoint subsets of approximately equal size. A series of k models are then fitted, each using a different combination of $k-1$ subsets. The model selection criterion (2) is then evaluated for each model, in each case using the subset of the data not used in fitting that model. The k-fold cross-validation estimate of the model selection criterion is then taken to be the mean of the criterion on the "test" data for each model. The most extreme form of cross-validation, in which each partition consists of a single pattern, is known as leave-one-out cross-validation [22].

5 Unbiased Estimation of the Predictive Variance

Maximum likelihood estimates of variance, whether homoscedastic or heteroscedastic are known to be biased. If over-fitting is present in the model of the predictive mean, then the apparent variance of training data is reduced as the model attempts to "explain" the realisation of the random noise process corrupting the data to some degree. This will cause any estimate of the conditional variance based on the predictive mean to be unrealistically low. For this reason, the conditional variance should be estimated using training samples which have not been used to estimate the conditional mean. In this study, we use instead a leave-one-out cross-validation estimate for the predictive variance. As a result, the model of the predictive variance is effectively fitted on data that has not been used to fit the model of the predictive mean, where in principle no over-fitting can have occurred and so the bias in the predictive variance is eliminated. This approach is equally valid for estimating the constant variance of a conventional kernel ridge regression model, for estimates of predictive variance made by a second kernel ridge regression model, or for the joint model of predictive mean and variance implemented by the heteroscedastic kernel ridge regression model. Fortunately, this approach is computationally feasible, as leave-one-out cross-validation can be performed efficiently in closed form for kernel learning methods minimising a (weighted) least-squares cost function [23].

5.1 Efficient Leave-One-Out Cross-Validation of Kernel Models

Consider a linear regression model $\hat{y}(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}) + b$ constructed in a feature space induced by a positive definite kernel, where the parameters $(\boldsymbol{w}, b)$ are given by the minimiser of a regularised weighted least-squares objective function,

$$L = \sum_{i=1}^{\ell} \lambda_i \left[y_i - \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) - b \right]^2 + \gamma \|\boldsymbol{w}\|^2.$$

The parameters of the resulting kernel expansion, $\hat{\boldsymbol{y}}(\boldsymbol{x}) = \sum_{i=1}^{\ell} \alpha_i \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b$, are given by the solution of a system of linear equations,

$$\begin{bmatrix} \boldsymbol{K} + \gamma \boldsymbol{\Lambda} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}\left\{\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_\ell^{-1}\right\}$. Let $\boldsymbol{H}$ represent the "hat" matrix, which maps the targets onto the model outputs, i.e. $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$, such that

$$\boldsymbol{H} = [h_{ij}]_{i,j=1}^{\ell} = [\boldsymbol{K}\ \mathbf{1}] \begin{bmatrix} \boldsymbol{K} + \gamma \boldsymbol{\Lambda} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} \tag{17}$$

For the sake of notational convenience, let $\hat{y}_j = \hat{y}(\boldsymbol{x}_j)$. During each iteration of the leave-one-out cross-validation procedure, a regression model is fitted using all but one of the available patterns. Let $\hat{y}_j^{(-i)}$ represent the output of the model for the j^{th} pattern during the i^{th} iteration of the leave-one-out procedure and $\hat{\boldsymbol{y}}^{(-i)} = \left(\hat{y}_1^{(-i)}, \hat{y}_2^{(-i)}, \ldots, \hat{y}_\ell^{(-i)}\right)$. Note that given any training set and the corresponding learned model, if one adds a point in the training set with target equal to the output predicted by the model, the model will not change since the cost function will not be increased by this new point. Here, given the training set with the point $\boldsymbol{x}_i$ left out, the predicted output are by definition $\hat{\boldsymbol{y}}^{(-i)}$ and they will not change if the point $\boldsymbol{x}_i$ is added with target $\hat{y}_i^{(-i)}$

$$\hat{\boldsymbol{y}}^{(-i)} = \boldsymbol{H}\boldsymbol{y}^*, \qquad \text{where} \qquad y_j^* = \begin{cases} y_j & \text{if } j \neq i \\ \hat{y}_j^{(-i)} & \text{if } j = i \end{cases}. \tag{18}$$

Subtracting y_i from both sides of the i^{th} equation in the system of linear equations (18),

$$\begin{aligned} \hat{y}_i^{(-i)} - y_i &= \sum_{j=1}^{\ell} h_{ij} y_j^* - y_i \\ &= \sum_{j \neq i} h_{ij} y_j + h_{ii} \hat{y}_i^{(-i)} - y_i \\ &= \sum_{j=1}^{\ell} h_{ij} y_j - y_i + h_{ii} \left\{ \hat{y}_i^{(-i)} - y_i \right\} \\ &= \hat{y}_i - y_i + h_{ii} \left\{ \hat{y}_i^{(-i)} - y_i \right\} \end{aligned}$$

This may be rearranged in order to obtain a closed form expression for the the residual for the i^{th} training pattern during the i^{th} iteration of the leave-one-out cross-validation procedure, e_i^{-i}, in terms of the residual for a model trained on the entire dataset for that pattern, e_i, and the i^{th} element of the principal diagonal of the hat matrix, h_{ii},

$$e_i^{(-i)} = y_i - \hat{y}_i^{(-i)} = \frac{y_i - \hat{y}_i}{1 - h_{ii}} = \frac{e_i}{1 - h_{ii}} \tag{19}$$

Note that the diagonal elements of the hat matrix lie in the range $[0,\ 1]$, and so the residuals under leave-one-out cross-validation can never be smaller in magnitude than the residuals for a model trained on the entire dataset. Therefore any estimate of predictive variance based on leave-one-out cross-validation will also be greater than the estimate based on the output of a model trained on the entire dataset, thereby reducing, if not actually eliminating, the known conservative bias in the latter. Another derivation of the leave-one-out error is given in Appendix A.

5.2 Experimental Demonstration

In this section we use a synthetic regression problem, taken from Williams [13], in which the true predictive standard deviation is known exactly, to demonstrate that the leave-one-out heteroscedastic kernel ridge regression (LOOHKRR) model provides almost unbiased estimates of the predictive standard deviation. The univariate input patterns, x, are drawn from a uniform distribution on the interval $(0,\ \pi)$; the corresponding targets, y, are drawn from a univariate Normal distribution with mean and variance that vary smoothly with x:

$$x \sim \mathcal{U}(0, \pi), \quad \text{and} \quad y \sim \mathcal{N}\left(\sin\left\{\frac{5x}{2}\right\}\sin\left\{\frac{3x}{2}\right\}, \frac{1}{100} + \frac{1}{4}\left[1 - \sin\left\{\frac{5x}{2}\right\}\right]^2\right).$$

Figure 1, parts (a) and (b), show the arithmetic mean of the predictive mean and $\pm$ one standard deviation credible interval for simple and leave-one-out heteroscedastic kernel ridge regression models respectively, over 1000 randomly generated realisations of the dataset, of 64 patterns each. A radial basis function kernel was used, with width parameter, $\kappa = 2$, for both the model of the predictive mean and the model of the predictive standard deviation, the regularisation parameters were set as follows: $\gamma^{\mu} = \gamma^{\sigma} = 1$ (the hyper-parameters we deliberately chosen to allow some over-fitting in the model of the predictive mean). In both cases the fitted mean is, on average, in good agreement with the true mean. Figure 1, parts (c) and (d), show the arithmetic mean of the predictive standard deviation for the simple and leave-one-out heteroscedastic kernel ridge regression models. The simple heteroscedastic kernel ridge regression model, on average, consistently under-estimates the conditional standard deviation, and so the predicted credible intervals are optimistically narrow. The mean predictive standard deviation for the leave-one-out heteroscedastic kernel ridge regression

model is very close to the true value. This suggests that the estimation of the predictive standard deviation is essentially unbiased as the expected value is approximately equal to the true value.

6 Gaussian Process Models

Gaussian Processes (GP) for regression [3] are powerful non parametric probabilistic models. They makes use of a prior covariance matrix of the targets $\boldsymbol{y}$ which has the form

$$K_{ij} = a(\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \gamma\delta_{ij}),$$

where $\mathcal{K}$ is any kernel function (for instance, the one defined in equation (3)), a is the *amplitude* parameter and γ is the *noise to signal ratio* parameter. Those parameters, as well as the hyper-parameters of the kernel are found by minimising the negative log *evidence*

$$\log\det \boldsymbol{K} + \boldsymbol{y}^\top \boldsymbol{K}^{-1}\boldsymbol{y}. \tag{20}$$

Note that a can be found in closed form,

$$a = \frac{\boldsymbol{y}^\top \boldsymbol{K}_{a=1}^{-1}\boldsymbol{y}}{n}.$$

The mean prediction is the same as in homoscedastic kernel ridge regression (without bias),

$$\mu(\boldsymbol{x}) = \boldsymbol{k}^\top(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{y},$$

with $\boldsymbol{k}^\top(\boldsymbol{x}) = a(\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, \mathcal{K}(\boldsymbol{x}_n, \boldsymbol{x}))$. The difference between kernel ridge regression and Gaussian Processes is that GP give a natural estimation of the predictive uncertainty as:

$$\sigma^2(\boldsymbol{x}) = a\gamma + a\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^\top(\boldsymbol{x})\boldsymbol{K}^{-1}\boldsymbol{k}(\boldsymbol{x}). \tag{21}$$

Note that the first term is constant and is the estimated noise level. The sum of the two others corresponds to the uncertainty in the mean prediction: for instance, it is large when the test point is far away from the training data.

Let us compare the leave-one-out predictive variances given by our method and by GP. For GP, if we let the point i out of the training set, its predictive variance will be:

$$a(K_{ii} - K_{\bar{i}i}^\top (K_{\bar{i}\bar{i}})^{-1} K_{\bar{i}i}) = \frac{a}{K_{ii}^{-1}} = \frac{y^\top K^{-1} y}{n K_{ii}^{-1}}, \tag{22}$$

where $K_{\bar{i}\bar{i}}$ is the matrix K with the i-th column and row removed. This is not completely exact as one should recompute a once the point i is out of the training set. But usually, values of hyper-parameters are not really affected by the leave-one-out procedure. For our method, the leave-one-out error on the point i is given by

$$\left(\frac{[K^{-1}y]_i}{K_{ii}^{-1}}\right)^2. \tag{23}$$

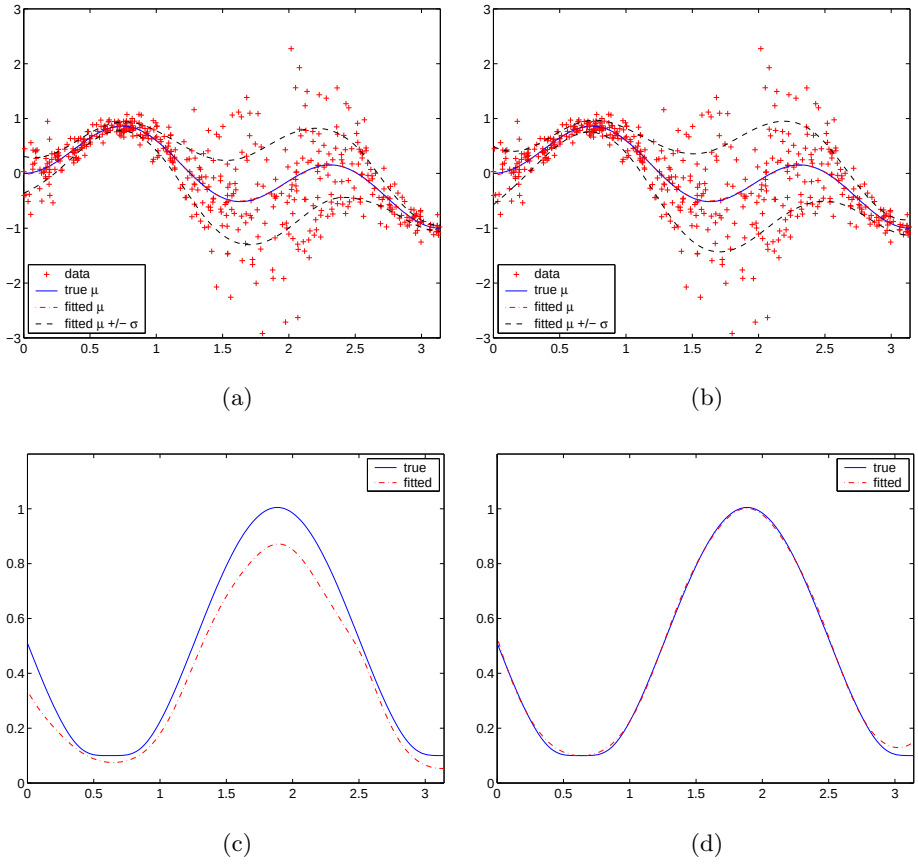

Fig. 1. Arithmetic mean of the estimate of the predictive mean and ± one standard deviation credible interval for (a) simple heteroscedastic kernel ridge regression (HKRR) and (b) leave-one-out heteroscedastic kernel ridge regression (LOOHKRR) models for a synthetic regression problem, (c) and (d) display the corresponding means of the estimated predictive standard deviation for the HKRR and LOOHKRR models respectively. All graphs show average results computed over 1000 randomly generated datasets (see text for details).

We can see that the two expressions are similar, but the GP takes the data less into account (the numerator is constant). This is not surprising, as in general, Bayesian methods rely more on the prior and less on the data. This yields near optimal predictions when the prior correctly reflects our knowledge of the problem, but can be suboptimal when there is prior mismatch. We will illustrate this point by the following toy problem. We want to model the step function on $[-1, 1]$, $f(t) = 1$ if $t > 0$, 0 otherwise. For this purpose, we used the Gaussian kernel (3). Note that this kernel is not the best suited for this task because it is smooth and stationary whereas the target function is not. The kernel width κ and the ridge γ have been optimised by minimising the negative log evidence (20). 100 points x_i have been chosen uniformly spaced in the interval $[-1, 1]$ and the targets have been corrupted with a Gaussian noise of standard deviation 0.1. The data and the mean prediction (which is the same for GP and kernel ridge regression) are plotted in the left of figure 2.

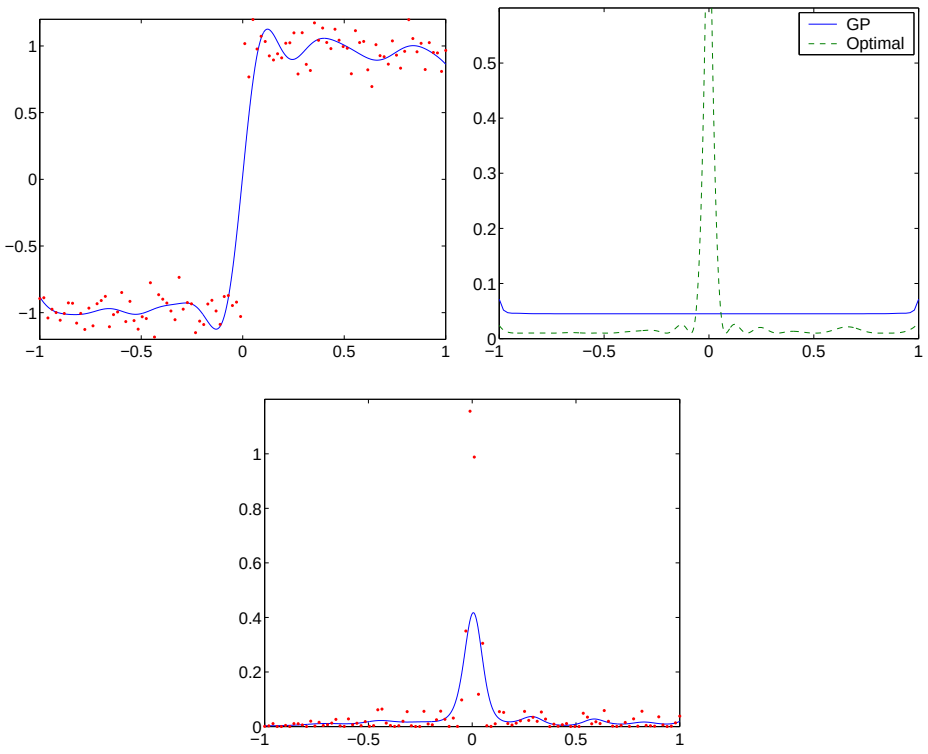

Fig. 2. Step function toy problem. *Left*: Training points and mean prediction. *Right:* GP predictive variance and the "optimal" one (given the mean). *Bottom*: Leave-one-out errors and the resulting predictive variance learned by the proposed method.

Given a test point x and the mean prediction $\mu(x)$, the "optimal" predictive variance (which we actually defined in the introduction as the conditional variance) is obtained by minimising the loss (2) and is

$$\sigma^2(x) = E_{y|x}(\mu(x) - y)^2 = (\mu(x) - f(x))^2 + \text{noise variance},$$

where $f(x)$ is the (unknown) target function. In our toy problem, we know the target function and the noise variance, so we can compute this optimal predictive uncertainty, as shown in the right of figure 2. We can see that this "optimal" predictive variance is very large around 0. This is because the mean prediction is not very good in this region and ideally, the predictive variance needs to be increased in regions where the mean prediction is far form the target function. However, when the kernel function used by the GP is stationary and the points are equally spaced, the predictive variance (21) given by the GP is almost constant, as shown in the right of figure 2: in this case, it is unable to see that the predictive variance should be increased around 0. The leave-one-out errors are plotted as dots in the bottom of figure 2. The first observation is that the misfit around 0 is well captured. However, the variance of the leave-one-out errors in the flat regions is high. This is directly related to the noise in the targets. For instance, it can happen that "by chance", the leave-one-out error on a given point is almost 0; but that does not mean that we are necessarily sure of the value of the function at this point. That is the reason why we have to perform a regression for the predictive variance (cf section 3.2). For this toy problem, we took the same kernel and regularisation parameter as used for the mean prediction and minimised (14), with ξ_i being the leave-one-out error on x_i. The estimated predicted variance is plotted at the bottom of figure 2. For this toy problem, the average negative log likelihoods (2) computed on a large test set are: -3.17 for the "optimal", -2.93 for our method and -2.3 for the GP. We would like to point out that in most real world examples, GP give reasonable predictive variances. This toy problem is just an illustration of what can happen in the case of a "prior mismatch" and how a non Bayesian method can overcome this difficulty.

As an aside, it is interesting to note that even if the leave-one-out predictive variance (22) for GP and the leave-one-out error (23) can be quite different, their *average* should be similar, as they are both estimate of the test error. On our toy problem, they were respectively 0.0447 and 0.0459, while the test error was 0.0359. We can try to see this similarity from an analytical point of view. First note that the gradient of (20) with respect to the ridge parameter should be 0, yielding

$$\text{trace } K^{-1} = \frac{1}{a} \sum [K^{-1} y]_i^2.$$

So the mean of (22) can be rewritten as

$$\frac{1}{n} \sum \frac{1}{K_{ii}^{-1}} \frac{\sum [K^{-1} y]_i^2}{\sum K_{ii}^{-1}},$$

which is very similar to the mean of (23),

$$\frac{1}{n}\sum\left(\frac{[K^{-1}y]_i}{K_{ii}^{-1}}\right)^2,$$

if the variance of the K_{ii}^{-1} is small.

7 Results for Challenge Benchmark Datasets

In this section, we detail results obtained on the three non-linear regression benchmark problems considered by the predictive uncertainty challenge, namely `gaze`, `stereopsis` and `Outaouais`. The methods that we considered are the following:

KRR Conventional kernel ridge regression with fixed variance prediction based on the training set MSE.

KRR + LOO Conventional kernel ridge regression with fixed variance prediction based on the leave-one-out estimate of the MSE.

KRR + KRR Conventional kernel ridge regression with predictive variance via kernel ridge regression on the residuals over the training set.

KRR + LOO + KRR Conventional kernel ridge regression with predictive variance via kernel ridge regression on the leave-one-out residuals.

HKRR Heteroscedastic kernel ridge regression.

LOOHKRR Heteroscedastic kernel ridge regression with unbiased estimation of the predictive variance. This is the method described in this paper.

7.1 Gaze

Table 1 shows the negative logarithm of the predictive density (NLPD) and mean squared error (MSE) for various kernel ridge regression-based models over training, validation and test partitions of the `gaze` benchmark dataset. A visual inspection of the data revealed that columns 3 and 4 of the validation and test partitions contained a small number of outliers (large negative values well outside the range of values observed in the training data). These outliers were "repaired" via a simple missing data imputation procedure based on linear regression. An isotropic Gaussian radial basis function kernels were used throughout, with model selection based on minimisation of the the 10-fold cross-validation estimate of the MSE (for standard kernel ridge regression models) or NLPD (for the heteroscedastic kernel ridge regression models). The use of leave-one-out cross-validation in fitting the model of the predictive variance also provides demonstrably better performance, with the KRR+LOO and KRR+LOO+KRR outperforming the KRR, and KRR+KRR models respectively. The very poor performance of the KRR+KRR model provides a graphic example of the dangers

associated with the unrealistically low estimates of predictive variance provided by existing approaches. In the case of the HKRR and LOOHKRR models, the NLPD is lower for the HKRR model because it provides a better model of the conditional mean. It should be noted, however, that the differences in test set NLPD between models, with the exception of the KRR and KRR+KRR, are generally very small and unlikely to be really meaningful.

Table 1. Performance of various models, based on kernel ridge regression, on the `gaze` dataset, in terms of mean squared error (MSE) and negative log predictive density (NLPD) over the training and validation partitions.

Mean description	**Train Set NLPD**	**Valid Set NLPD**	**Test Set NLPD**	**Train Set MSE**	**Valid Set MSE**	**Test Set MSE**
KRR	4.723	5.776	5.8172	0.01165	0.03654	0.04029
KRR+LOO	4.912	5.292	5.3077	0.01165	0.03653	0.04029
KRR+KRR	5.003	12.119	7.6011	0.01165	0.03653	0.04029
KRR+LOO+KRR	4.857	5.282	5.2951	0.01165	0.03653	0.04029
HKRR	5.119	5.248	5.2650	0.02574	0.03272	0.03607
LOOHKRR	4.881	5.305	5.3214	0.01159	0.03677	0.04051

7.2 Stereopsis

Table 2 shows the negative logarithm of the predictive density (NLPD) and mean squared error (MSE) for various kernel ridge regression models over training, validation and test partitions of of the `stereopsis` benchmark dataset. An anisotropic Gaussian radial basis function kernels and model selection based on validation set NLPD are used throughout. The labels for the first six models are as described for the `gaze` dataset. An investigation of the test data revealed that the negative log-likelihood for one of the test patterns dominated the contribution from the other patterns, as shown in Figure 3 (a). An advantage of generating a predictive distribution, rather than a single point prediction, is that it is possible to detect potential outliers in the test data (i.e. observations that cannot be reconciled with an otherwise accurate model of the data). If we choose to interpret the results as indicating, for instance a data entry error, and delete pattern number 162, the resulting test-set NLPD statistics are much more closely in accord with the corresponding validation set statistics. Looking at the data in more detail, we can see that the test targets are clustered into 10 relatively compact clusters. Pattern #162 belongs to the cluster of values lying between 150 and 170, Figure 3 (b). Figure 3 (c) the projection of points with targets lying between 150 and 170 onto the first two principal components of the corresponding input features (excluding pattern #162). This shows that the input features for pattern #162 are atypical of patterns with a target of

≈ 160. Figure 3 (d) shows the results obtained using simple linear regression on all patterns belonging to this cluster, excluding pattern #162 (blue circles). It can be seen that there is a reasonably strong correlation between the predicted and true target values. The prediction of this model on pattern #162 predicts a much lower target value, suggesting that the relationship between target and input features for pattern #162 is different than that for the rest of the cluster. The predicted targets for a KRR model based on the entire training partition are also shown (green $\times$ and black square). Again the model predicts a target significantly lower than the given target value. This suggests the model may well be correct in assigning a very low likelihood to pattern #162.

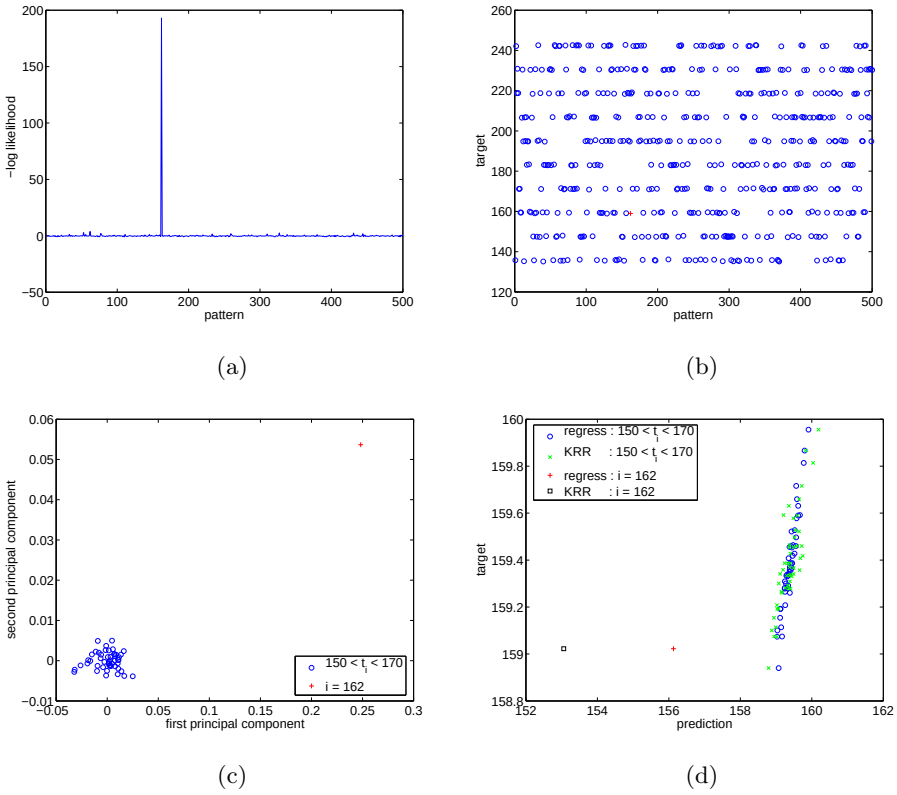

Fig. 3. Analysis of `stereopsis` dataset: (a) The negative log-likelihood is dominated by the contribution from pattern #162. (b) Illustration of the discrete nature of the test targets. (c) Plot of the projection of points with targets lying between 150 and 170 onto the first two principal components of the corresponding input features. (d) Regression results demonstrate that pattern #162 is clearly an outlier.

The results for the `stereopsis` dataset are more equivocal than those for the `gaze` dataset. Again, a modest improvement in validation and test set NLPD is

obtained through the use of leave-one-out cross-validation in fitting the model of the conditional variance, in the case of KRR/KRR+LOO and KRR+KRR/ KRR+LOO+KRR models. However in this case, both HKRR and LOOHKRR models perform poorly. This may be because the data were not collected randomly across the pattern space and this complicates the regularisation of the model.

Table 2. Performance of various models, based on kernel ridge regression, on the `stereopsis` dataset, in terms of mean squared error (MSE) and negative log predictive density (NLPD) over the training and validation partitions. Two values of the NLPD for the test set are given; the first gives the NLPD computed over the entire test set, the second excludes the problematic pattern #162.

Model description	**Train Set NLPD**	**Valid Set NLPD**	**Test Set NLPD 1**	**Test Set NLPD 2**	**Train Set MSE**	**Valid Set MSE**	**Test Set MSE**
KRR	-0.5930	+0.0241	+1.8742	-0.1124	1.464×10^5	3.481×10^5	3.095×10^5
KRR+LOO	-0.4917	-0.1889	+0.7189	-0.2559	1.464×10^5	3.481×10^5	3.095×10^5
KRR+KRR	-0.6194	+0.0620	+1.4088	-0.0805	1.464×10^5	3.481×10^5	3.095×10^5
KRR+LOO+KRR	-0.5835	-0.2459	+0.4924	-0.2718	1.464×10^5	3.481×10^5	3.095×10^5
HKRR	-0.3940	-0.2061	+1.6928	-0.1306	2.176×10^5	3.041×10^5	3.369×10^5
LOOHKRR	-0.4813	-0.1798	+2.8873	-0.0803	1.725×10^5	2.599×10^5	2.860×10^5
KRR + Quant. Var.	-0.2726	-0.0872	+0.2626	-0.1238	1.288×10^5	3.892×10^5	3.447×10^5
KRR Mixture	-2.3967	-1.5538	+121.00	-1.6173	0.025×10^5	0.169×10^5	1.681×10^5

The last two rows of Table 2 relate to further experiments inspired by the solution of Snelson and Murray, who noticed that the targets for this dataset were strongly clustered into ten compact groups. The KRR + Quant. Var. model adopted a KRR model of the predictive mean, and then estimated the constant variance separately for each cluster. The KRR mixture model used a KRR model to estimate the predictive mean of the target distribution and one of a set of ten KRR models used to estimate the predictive variance within each cluster, depending on the estimate of the predictive mean. The KRR Mixture model clearly provides a substantial improvement in the achievable validation set NLPD. However the clustering of the target values was later revealed to be an artifact of the data collection process, and so this improvement is essentially meaningless as this approach would not be feasible in operation.

7.3 Outaouais

The `outaouais` dataset is the largest of the challenge benchmarks, and is too large ($20,000$ training patterns and 37 features) to easily apply kernel learning methods directly. We therefore modelled this dataset using a multi-layer perceptron network (e.g. [9]), with a heteroscedastic loss function [13] similar to that used in training the heteroscedastic kernel ridge regression model. Bayesian regularisation with a Laplace prior [24] was used to avoid over-fitting the training data and to identify and prune redundant connections. It is interesting to

note that this, rather dated, technique performed quite creditably, as shown in Table 3.

Table 3. Performance of various models on the `outaouais` dataset, in terms of mean squared error (MSE) and negative log predictive density (NLPD) over the training, validation and test partitions. All the numbers are multiplied by 100.

Model description	Train NLPD	Test NLPD	Valid NLPD	Train MSE	Test MSE	Valid MSE
Gaussian process	-92.55	-92.13	-92.55	0	1.727	0
Classification + NN	-152.4	-87.95	-152.4	0	5.635	0
CAN + CV	-86.68	-64.81	-87.59	1.784	3.774	1.636
Heteroscedastic MLP	-32.99	-23.04	-22.15	19.55	20.13	19.27
Gaussian Process	3.246	9.019	11.79	14.9	15.8	16.48
MDN Ensemble	17.93	19.93	19.56	27.72	27.83	27.99
NeuralBAG/EANN	47.68	50.5	49.44	26.71	27.03	26.63
baseline	109.5	111.5	112.4	10	10	10

8 Conclusions

In this paper, we have shown that the assumption of a *heteroscedastic* (input dependent) noise structure can improve the performance of kernel learning methods for non-linear regression problems. The resulting estimate of the predictive variance provides a useful estimate of the uncertainty inherent in the usual estimate of the predictive mean. We have also demonstrated that leave-one-out cross-validation, which can be implemented very efficiently in closed form for a variety of kernel learning algorithms, can be used to overcome the bias inherent in (penalised) maximum likelihood estimates of predictive variance. It would be interesting to compare the leave-one-out cross-validation method investigated here with the Bayesian scheme proposed by Bishop and Qazaz [25], which instead marginalises over the estimate of the predictive mean in fitting the model of the predictive variance, or the Gaussian process treatment of Goldberg *et al.* [26].

Appendix A: an alternative derivation of the leave-one-out error

We present here an other derivation of (19). Suppose that the point $\boldsymbol{x}_1$ is taken out of the training set. Let $\boldsymbol{\alpha}^{(-1)}$ and $b^{(-1)}$ the parameters found by kernel ridge regression and let us write the following block matrix decomposition:

$$\begin{bmatrix} \boldsymbol{K} + \gamma \boldsymbol{\Lambda} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} = \begin{bmatrix} m_{11} & \boldsymbol{m_1}^\top \\ \boldsymbol{m_1} & \boldsymbol{M_1} \end{bmatrix} \equiv \boldsymbol{M}$$

Then

$$\begin{bmatrix} \boldsymbol{\alpha}^{(-1)} \\ b^{(-1)} \end{bmatrix} = {\boldsymbol{M_1}}^{-1}[y_2 \dots y_n \ 0]^\top$$

And

$$\begin{aligned} \hat{y}_1^{(-1)} &= \boldsymbol{m_1}^\top [\boldsymbol{\alpha}^{(-1)} \ b^{(-1)}]^\top \\ &= \boldsymbol{m_1}^\top {\boldsymbol{M_1}}^{-1}[y_2 \dots y_n \ 0]^\top \\ &= \boldsymbol{m_1}^\top {\boldsymbol{M_1}}^{-1}[\boldsymbol{m_1} \ \boldsymbol{M_1}][\boldsymbol{\alpha} \ b]^\top \\ &= \boldsymbol{m_1}^\top {\boldsymbol{M_1}}^{-1}\boldsymbol{m_1} \ \alpha_1 + \boldsymbol{m_1}^\top [\alpha_2 \cdots \alpha_n \ b]^\top \end{aligned}$$

On the other hand, the first row of the vector equality $\boldsymbol{M}[\boldsymbol{\alpha} \ b]^\top = \boldsymbol{y}$ gives $y_1 = m_{11}\alpha_1 + \boldsymbol{m_1}^\top [\alpha_2 \cdots \alpha_n \ b]^\top$. And thus we get

$$\begin{aligned} y_1 - \hat{y}_1^{(-1)} &= \alpha_1 (m_{11} - \boldsymbol{m_1}^\top {\boldsymbol{M_1}}^{-1}\boldsymbol{m_1}) \\ &= \frac{\alpha_1}{(\boldsymbol{M}^{-1})_{11}} \end{aligned} \tag{24}$$

The last equality comes from block matrix inversion (also known as Schur complement). Thus computing the leave-one-out error only requires the inversion of the matrix $\boldsymbol{M}$ (and this matrix has been previously inverted to find the coefficients $\boldsymbol{\alpha}$ and b of the kernel ridge regression algorithm).

This result is the same as (19). Indeed, the denominator $1 - h_{ii}$ is the i-th diagonal element of

$$I - \begin{bmatrix} \boldsymbol{K} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \boldsymbol{M}^{-1} = \left(\boldsymbol{M} - \begin{bmatrix} \boldsymbol{K} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \right) \boldsymbol{M}^{-1} = \gamma \boldsymbol{\Lambda} \boldsymbol{M}^{-1}.$$

The first equality comes from the definition of $\boldsymbol{H}$ (17). Finally, combining with (12) (with $\lambda_i = \lambda_i$ and $\gamma^\mu = \gamma$), we get

$$\frac{e_i}{1 - h_{ii}} = \frac{e_i \lambda_i}{\gamma (\boldsymbol{M}^{-1})_{ii}} = \frac{\alpha_i}{(\boldsymbol{M}^{-i})_{ii}}.$$

Note that even though (19) and (24) are equal, the latter might be more numerically stable when γ is very small: indeed, in this case $h_{ii} \approx 1$.

References

1. Candela, J.Q.: Evaluating predictive uncertainty challenge (2005) predict.kyb.tuebingen.mpg.de.
2. Saunders, C., Gammerman, A., Vovk, V.: Ridge regression learning algorithm in dual variables. In: Proc., 15th Int. Conf. on Machine Learning, Madison, WI (1998) 515–521
3. Williams, C., Rasmussen, C.: Gaussian Processes for Regression. In Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems, NIPS. Volume 8., MIT Press (1995)

4. Suykens, J.A.K., De Brabanter, J., Lukas, L., Vanderwalle, J.: Weighted least squares support vector machines : robustness and sparse approximation. Neurocomputing **48** (2002) 85–105
5. Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society of London, A **209** (1909) 415–446
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines (and other kernel-based learning methods). Cambridge University Press, Cambridge, U.K. (2000)
7. Schölkopf, B., Smola, A.J.: Learning with kernels - support vector machines, regularization, optimization and beyond. MIT Press, Cambridge, MA (2002)
8. Hoerl, A.E., Kennard, R.W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics **12** (1970) 55–67
9. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
10. Satchwell, C.: Finding error bars (the easy way). Neural Computing Applications Forum **5** (1994)
11. Lowe, D., Zapart, C.: Point-wise confidence interval estimation by neural networks: A comparative study based on automotive engine calibration. Neural Computing and Applications **8** (1999) 77–85
12. Nix, D.A., Weigend, A.S.: Estimating the mean and variance of the target probability distribution. In: Proceedings of the IEEE International Conference on Neural Networks. Volume 1., Orlando, FL (1994) 55–60
13. Williams, P.M.: Using neural networks to model conditional multivariate densities. Neural Computation **8** (1996) 843–854
14. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons (1998)
15. Kimeldorf, G.S., Wahba, G.: Some results on Tchebycheffian spline functions. Journal of Mathematical Analysis and Applications **33** (1971) 82–95
16. Schölkopf, B., Herbrich, R., Smola, A.J.: A generalised representer theorem. In: Proceedings of the Fourteenth International Conference on Computational Learning Theory, Amsterdam, the Netherlands (2001) 416–426
17. Cawley, G.C., Talbot, N.L.C., Foxall, R.J., Dorling, S.R., Mandic, D.P.: Heteroscedastic kernel ridge regression. Neurocomputing **57** (2004) 105–124
18. Foxall, R.J., Cawley, G.C., Talbot, N.L.C., Dorling, S.R., Mandic, D.P.: Heteroscedastic regularised kernel regression for prediction of episodes of poor air quality. In: Proceedings of the European Symposium on Artificial Neural Networks (ESANN-2002), Bruges, Belgium (2002) 19–24
19. Yuan, M., Wahba, G.: Doubly penalized likelihood estimator in heteroscedastic regression. Statistics and Probability Letters **69** (2004) 11–20
20. Nabney, I.T.: Efficient training of RBF networks for classification. In: Proceedings of the Ninth International Conference on Artificial Neural Networks. Volume 1., Edinburgh, United Kingdom (1999) 210–215
21. Stone, M.: Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society, B **36** (1974) 111–147
22. Luntz, A., Brailovsky, V.: On estimation of characters obtained in statistical procedure of recognition (in Russian). Techicheskaya Kibernetica **3** (1969)
23. Cawley, G.C., Talbot, N.L.C.: Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. Pattern Recognition **36** (2003) 2585–2592
24. Williams, P.M.: Bayesian regularization and pruning using a Laplace prior. Neural Computation **7** (1995) 117–143

25. Bishop, C.M., Qazaz, C.S.: Bayesian inference of noise levels in regression. In von der Malsburg, C., von Seelen, W., Vorbrüggen, J.C., Sendhoff, B., eds.: Proceedings of the International Conference on Artificial Neural Networks (ICANN-96). Volume 1112 of Lecture Notes in Computer Science., Bochum, Germany, Springer (1996) 59–64
26. Goldberg, P.W., Williams, C.K.I., Bishop, C.M.: Regression with input-dependent noise : A Gaussian process treatment. In Jordan, M., Kearns, M., Solla, S., eds.: Advances in Neural Information Processing Systems. Volume 10. MIT Press (1998) 493–499