AARHUS
UNIVERSITY
BUSINESS AND SOCIAL SCIENCES
DEPARTMENT OF ECONOMICS AND BUSINESS

CREATES
Center for Research in Econometric Analysis of Time Series

Selection in Kernel Ridge Regression

Peter Exterkate

CREATES Research Paper 2012-10

Department of Economics and Business
Aarhus University
Bartholins Allé 10
DK-8000 Aarhus C
Denmark

Email: oekonomi@au.dk
Tel: +45 8716 5515

Model Selection in Kernel Ridge Regression*

Peter Exterkate†

CREATES, Aarhus University

February 28, 2012

Abstract

Kernel ridge regression is gaining popularity as a data-rich nonlinear forecasting tool, which is applicable in many different contexts. This paper investigates the influence of the choice of kernel and the setting of tuning parameters on forecast accuracy. We review several popular kernels, including polynomial kernels, the Gaussian kernel, and the Sinc kernel. We interpret the latter two kernels in terms of their smoothing properties, and we relate the tuning parameters associated to all these kernels to smoothness measures of the prediction function and to the signal-to-noise ratio. Based on these interpretations, we provide guidelines for selecting the tuning parameters from small grids using cross-validation. A Monte Carlo study confirms the practical usefulness of these rules of thumb. Finally, the flexible and smooth functional forms provided by the Gaussian and Sinc kernels makes them widely applicable, and we recommend their use instead of the popular polynomial kernels in general settings, in which no information on the data-generating process is available.

Keywords: Nonlinear forecasting, shrinkage estimation, kernel methods, high dimensionality.
JEL Classification: C51, C53, C63.

*The author would like to thank Christiaan Heij, Patrick Groenen, and Dick van Dijk for useful comments and suggestions. The author acknowledges support from CREATES, funded by the Danish National Research Foundation.

†Address for correspondence: CREATES, Department of Economics and Business, Aarhus University, Bartholins Allé 10, 8000 Aarhus C, Denmark; email: exterkate@creates.au.dk; phone: +45 8716 5548.

1 Introduction

In many areas of application, forecasters face a trade-off between model complexity and forecast accuracy. Due to the uncertainty associated with model choice and parameter estimation, a highly complex nonlinear predictive model is often found to produce less accurate forecasts than a simpler, e.g. linear, model. Thus, a researcher wishing to estimate a nonlinear relation for forecasting purposes generally restricts the search space drastically, for example to polynomials of low degree, or to regime-switching models (Teräsvirta, 2006) or neural networks (White, 2006). A recent comprehensive overview was given by Kock and Teräsvirta (2011). The improvement of such models upon the predictive accuracy of linear models is often found to be limited, see for example Stock and Watson (1999), Teräsvirta et al. (2005), and Medeiros et al. (2006).

Another manifestation of this complexity-accuracy trade-off is that, while a very large number of potentially relevant predictors may be available, the *curse of dimensionality* implies that better forecasts can be obtained if a large proportion of the predictors is discarded. This situation arises, for example, in economic applications. Hundreds or even thousands of predictors are often available, and economic theory does not usually provide guidelines concerning which variables should or should not influence each other. A reduction in the number of predictors can of course be achieved by selecting a small subset of representative variables, but the most common way to proceed is to summarize the predictors by a small number of principal components. This approach has found successful forecasting applications in macroeconomics (e.g. Stock and Watson, 2002) and in finance (e.g. Ludvigson and Ng, 2007, 2009).

In this paper we discuss *kernel ridge regression*, a forecasting technique that can overcome both aspects of this trade-off simultaneously, making it suitable for estimating nonlinear models with many predictors. While kernel methods are not widely known in the fields of economics and finance, they have found ample applications in machine learning; a recent review can be found in Hofmann et al. (2008). A typical application is classification, such as optical recognition of handwritten characters (Schölkopf et al., 1998). Recently, Exterkate et al. (2011) use this technique in a macroeconomic forecasting application and they report an increase in forecast accuracy, compared to traditional linear methods.

The central idea in kernel ridge regression is to employ a flexible set of nonlinear prediction functions and to prevent overfitting by penalization, in a way that limits the computational complexity. This is achieved by mapping the set of predictors into a high-dimensional (or even infinite-dimensional)

space of nonlinear functions of the predictors. A linear forecast equation is then estimated in this high-dimensional space, using a penalty (or shrinkage, or ridge) term to avoid overfitting. Computational tractability is achieved by choosing the mapping in a convenient way, so that calculations in the high-dimensional space are actually prevented.

Kernel ridge regression provides the practitioner with a large amount of flexibility, but it also leaves him with a number of nontrivial decisions to make. One such decision concerns which kernel to use. Although any choice of kernel leads to restrictions on the functional form of the forecast equation, little attention is generally paid to such implications. Additionally, kernel ridge regression involves tuning parameters, and their practical interpretation is not always clear. This feature makes it difficult to select "reasonable" values for these parameters, resulting in time-consuming grid searches or in suboptimal forecasting performance.

To give a clear interpretation of the kernel functions and their associated tuning parameters, we review the kernel methodology from two different points of view, namely, function approximation and Bayesian statistics. This combination of perspectives enables us to relate one of the two tuning parameters that are found in most applications of kernel ridge regression to the signal-to-noise ratio in the data, and the other to smoothness measures of the prediction function. Based on these insights, we give explicit rules of thumb for selecting their values by using cross-validation over small grids. Cross-validation may also be used to select among different types of kernel. However, one needs to be somewhat careful with this procedure: we provide empirical evidence against including the popular polynomial kernels in the cross-validation exercise.

In Section 2 we describe the kernel methodology, from the perspective of function approximation and from Bayesian statistics. We discuss several popular kernels and the functional forms of their associated forecast equations, and we interpret their tuning parameters. Section 3 presents a Monte Carlo simulation to show the effects of different methods for choosing the kernel and its tuning parameters. Concerning the tuning parameters, selecting them using cross-validation from our grids affects the forecast quality only marginally, compared to using the true values. The choice of kernel can also be left to cross-validation; however, using a polynomial kernel when the data-generating process is non-polynomial reduces forecast accuracy. We also present simulations in which all kernels estimate misspecified models, and we find that the "smooth" Gaussian and Sinc kernels outperform polynomial kernels in this case. We provide conclusions in Section 4.

2 Methodology

Kernel ridge regression can be understood as a function approximation tool, but it can also be given a Bayesian interpretation. We review the method from both viewpoints in Sections 2.1 and 2.2, respectively. We present some popular kernel functions in Section 2.3. In Section 2.4 we give an interpretation to the associated tuning parameters, and we derive "reasonable" values for these parameters.

2.1 Kernel ridge regression for function approximation

We first introduce some notation. We are given T observations $(y_1, x_1), (y_2, x_2), \ldots, (y_T, x_T)$, with $y_t \in \mathbb{R}$ and $x_t \in \mathbb{R}^N$, and our goal is to find a function f so that $f(x_t)$ is a "good" approximation to y_t for all $t = 1, 2, \ldots, T$. Then, we are given a new observation $x_* \in \mathbb{R}^N$ and asked to predict the corresponding y_*. We denote this prediction by $\hat{y}_* = f(x_*)$. By selecting f from a large and flexible class of functions while preventing overfitting, we hope to achieve that this prediction is accurate.

To describe the class of functions from which we select f, we first choose a mapping $\varphi : \mathbb{R}^N \to \mathbb{R}^M$. The regression function f will be restricted to a certain set of linear combinations of the form $\varphi(x)' \gamma$, with coefficient vector $\gamma \in \mathbb{R}^M$. The number of regressors M is either a finite integer with $M \geq N$, or $M = \mathbb{N}$, representing a countably infinite number of regressors. Examples of mappings of both types are presented in Section 2.3 below.

If a flexible functional form is desired, the number of regressors M needs to be large. Therefore we wish to avoid M-dimensional computations, and it turns out that we can do so by requiring only that the dot product $\kappa(x_s, x_t) = \varphi(x_s)' \varphi(x_t)$ can be found using only N-dimensional computations, for any $x_s, x_t \in \mathbb{R}^N$. In the machine learning literature this idea is known as the *kernel trick* (Boser et al., 1992), and the function $\kappa : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is commonly called the kernel function. Conversely, functions κ for which a corresponding φ exists can be characterized by a set of conditions due to Mercer (1909). All kernel functions discussed in this study satisfy these conditions; a thorough justification can be found in Hofmann et al. (2008).

Finally, define a space of functions $\mathcal{H}_0$ consisting of the functions $f : \mathbb{R}^N \to \mathbb{R}$ of the form $f(x) = \sum_{s=1}^{S} \alpha_s^f \kappa\left(x, x_s^f\right)$, for a finite set $x_1^f, x_2^f, \ldots, x_S^f \in \mathbb{R}^N$ and real numbers $\alpha_1^f, \alpha_2^f, \ldots, \alpha_S^f$. Every such $f(x)$ is a linear combination of the elements of $\varphi(x)$, as can be seen by recalling the definition of κ: we have $f(x) = \varphi(x)' \left(\sum_{s=1}^{S} \alpha_s^f \varphi\left(x_s^f\right)\right)$. We equip $\mathcal{H}_0$ with the following dot product:

if $f(x) = \sum_{s=1}^{S} \alpha_s^f \kappa\left(x, x_s^f\right)$ and $g(x) = \sum_{s'=1}^{S'} \alpha_{s'} \kappa\left(x, x_{s'}^g\right)$, then $\langle f, g \rangle_{\mathcal{H}} = \sum_{s=1}^{S} \sum_{s'=1}^{S'} \alpha_s^f \alpha_{s'}^g \kappa\left(x_s^f, x_{s'}^g\right)$.

(For the verification that $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is indeed a valid dot product, see Hofmann et al. (2008).) Finally, Aronszajn (1950) proved that completing $\mathcal{H}_0$ in the corresponding norm $||f||_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$ leads to a Hilbert space, which we call $\mathcal{H}$. This is the class of functions from which f will be selected.

In finite samples, an unrestricted search over the space $\mathcal{H}$ will lead to overfitting. Indeed, if $\mathcal{H}$ allows for sufficiently flexible functional forms, a prediction function f may be obtained for which the in-sample fit is perfect, but the out-of-sample predictive accuracy will generally be poor. Therefore, we consider the regularized problem

$$\min_{f \in \mathcal{H}} \sum_{t=1}^{T} (y_t - f(x_t))^2 + \lambda ||f||_{\mathcal{H}}^2, \tag{1}$$

for some $\lambda > 0$. A result due to Kimeldorf and Wahba (1971), known as the *representer theorem*, states that the minimizer of this problem can be written as $f(x) = \sum_{t=1}^{T} \alpha_t \kappa(x, x_t)$, for some sequence of real numbers $\alpha_1, \alpha_2, \ldots, \alpha_T$. That is, the optimal prediction function admits a kernel expansion in terms of the observations: the set of expansion points $\left\{x_1^f, x_2^f, \ldots, x_S^f\right\}$ may be taken equal to $\{x_1, x_2, \ldots, x_T\}$.

If we define $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{pmatrix}$, $\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_T \end{pmatrix}$, and $K = \begin{pmatrix} \kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots & \kappa(x_1, x_T) \\ \kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots & \kappa(x_2, x_T) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(x_T, x_1) & \kappa(x_T, x_2) & \cdots & \kappa(x_T, x_T) \end{pmatrix}$,

we see that problem (1) is equivalent to

$$\min_{\alpha \in \mathbb{R}^T} (y - K\alpha)'(y - K\alpha) + \lambda \alpha' K \alpha. \tag{2}$$

Minimizing the quadratic form in (2) yields $\alpha = (K + \lambda I)^{-1} y$, where I is the $T \times T$ identity matrix. Finally, to forecast a new observation y_* given the corresponding x_*, we have

$$\hat{y}_* = f(x_*) = \sum_{t=1}^{T} \alpha_t \kappa(x_*, x_t) = k_*' \alpha = k_*' (K + \lambda I)^{-1} y, \tag{3}$$

where the vector $k_* \in \mathbb{R}^T$ has $\kappa(x_*, x_t)$ as its t-th element. Note that (3) implies that the forecast $\hat{y}_*$ can be calculated without any computations in M-dimensional space, as desired.

2.2 Kernel ridge regression for Bayesian prediction

In this section we retain the notation introduced above, but our point of view is different. We assume that, conditional on x_t, each y_t has a normal distribution, with mean $\varphi(x_t)'\gamma$ for some $\gamma \in \mathbb{R}^M$, and with fixed but unknown variance θ^2. If we let Z be the $T \times M$ matrix[1] with t-th row equal to $\varphi(x_t)'$, the probability density function may be written as

$$p\left(y|Z,\gamma,\theta^2\right) \propto \left(\theta^2\right)^{-T/2} \exp\left(\frac{-1}{2\theta^2}(y-Z\gamma)'(y-Z\gamma)\right).$$

We specify our prior beliefs about γ and θ^2 as follows. We take the uninformative Jeffreys prior on θ^2 and, given θ^2, our prior on the distribution of γ is normal with mean zero and variance $\left(\theta^2/\lambda\right) I$:

$$p\left(\theta^2\right) \propto \left(\theta^2\right)^{-1}, \qquad p\left(\gamma|\theta^2\right) \propto \left(\theta^2\right)^{-M/2} \exp\left(\frac{-\lambda}{2\theta^2}\gamma'\gamma\right).$$

Using Bayes' rule, the posterior density of the parameters is given by

$$\begin{aligned} p\left(\gamma,\theta^2|Z,y\right) &\propto p\left(y|Z,\gamma,\theta^2\right) p\left(\gamma|\theta^2\right) p\left(\theta^2\right) \\ &\propto \left(\theta^2\right)^{-(T+M+2)/2} \exp\left(\frac{-1}{2\theta^2}\left[(y-Z\gamma)'(y-Z\gamma)+\lambda\gamma'\gamma\right]\right), \end{aligned}$$

see e.g. Raiffa and Schlaifer (1961). Now, for a new observation $x_* \in \mathbb{R}^N$, denote $z_* = \varphi(x_*)$ and assume that, just like $y_1, y_2, \ldots, y_T$, the unobserved y_* follows the normal distribution

$$p\left(y_*|z_*,\gamma,\theta^2,Z,y\right) \propto \left(\theta^2\right)^{-1/2} \exp\left(\frac{-1}{2\theta^2}\left(y_*-z_*'\gamma\right)^2\right).$$

Then, again by Bayes' rule, the predictive density of y_*, given all observed data, is

$$\begin{aligned} p\left(y_*|z_*,Z,y\right) &= \int_{\mathbb{R}^M}\int_0^\infty p\left(y_*|z_*,\gamma,\theta^2,Z,y\right) p\left(\gamma,\theta^2|Z,y\right) d\theta^2 d\gamma \\ &= \int_{\mathbb{R}^M}\int_0^\infty \left(\theta^2\right)^{-(T+M+3)/2} \\ &\qquad \times \exp\left(\frac{-1}{2\theta^2}\left[(y-Z\gamma)'(y-Z\gamma)+\left(y_*-z_*'\gamma\right)^2+\lambda\gamma'\gamma\right]\right) d\theta^2 d\gamma. \end{aligned}$$

[1] If M is infinite, applying the derivations in this section to a finite subset of the regressors and then letting $M \to \infty$ leads to the same final results.

This integral can be evaluated analytically (see e.g. Raiffa and Schlaifer, 1961). The resulting predictive density has $\hat{y}_*$ from (3) as its mean, median, and mode. More precisely, introducing $k_{**} = z_*' z_*$ and

$$w = \frac{1}{T} y' \left(K + \lambda I\right)^{-1} y \left(k_{**} + \lambda - k_*' \left(K + \lambda I\right)^{-1} k_*\right),$$

the quantity $w^{-1/2} \left(y_* - \hat{y}_*\right)$ follows Student's t distribution with T degrees of freedom.

That is, two different approaches to forecasting y_* in terms of linear combinations of certain functions of x_* yield the same point forecast $\hat{y}_*$. We exploit both points of view in the next section, which describes some common kernel functions, and in Section 2.4, where we discuss the associated tuning parameters.

2.3 Some popular kernel functions

A first obvious way of introducing nonlinearity in the prediction function $f\left(x\right) = \varphi\left(x\right)' \gamma$ is by making it a polynomial of some specified degree d. That is, we choose φ in such a way that $\varphi\left(x\right)$ contains all $\begin{pmatrix} N+d \\ d \end{pmatrix}$ monomials of the form $x_1^{d_1} x_2^{d_2} \cdots x_N^{d_N}$, with all d_n nonnegative integers with $\sum_{n=1}^N d_n \leq d$. As shown by Poggio (1975), the kernel function takes a simple form if we multiply each monomial by a constant: if a typical element of $\varphi\left(x\right)$ is

$$\left(\sigma^{-\sum_{n=1}^N d_n}\right) \sqrt{\frac{d!}{\left(d - \sum_{n=1}^N d_n\right)! \prod_{n=1}^N d_n!}} \prod_{n=1}^N x_n^{d_n}, \tag{4}$$

where $\sigma > 0$ is a tuning parameter, then the kernel function is simply

$$\kappa\left(x_s, x_t\right) = \left(1 + \frac{x_s' x_t}{\sigma^2}\right)^d. \tag{5}$$

As desired, this expression can be computed without constructing all the regressors of the form (4), which enables fast computation of forecasts as discussed in Section 2.1.

A more sophisticated method for constructing kernels is to require that the resulting prediction function must be smooth in some sense. From the point of view of function approximation, this is a sensible requirement, as we do not want to overfit the data. In the context of Section 2.1, we can achieve this by selecting κ to generate a Hilbert space $\mathcal{H}$ for which $||f||_{\mathcal{H}}$ measures lack of smoothness of f, as can be seen from the objective (1).

Following Smola et al. (1998), we restrict ourselves to functions f for which $\int_{\mathbb{R}^N} f(x)^2 dx$ is finite, and we measure the smoothness of such a function by examining its Fourier transform, defined by

$$\tilde{f} : \mathbb{R}^N \to \mathbb{R} \quad \text{with} \quad \tilde{f}(\omega) = (2\pi)^{-N/2} \int_{\mathbb{R}^N} \exp\left(-i\omega' x\right) f(x)\, dx.$$

The Fourier transform decomposes f according to frequency. That is, if $\tilde{f}(\omega)$ takes large values for large values of $||\omega||$, this indicates that $f(x)$ fluctuates rapidly with x, i.e., that f is not smooth. It follows that lack of smoothness of f can be penalized by choosing κ in such a way that

$$||f||_{\mathcal{H}} = (2\pi)^{-N} \int_{\mathbb{R}^N} \frac{\left|\tilde{f}(\omega)\right|^2}{v(\omega)} d\omega, \tag{6}$$

where $(2\pi)^{-N}$ is a normalization constant, $|\cdot|$ denotes the absolute value of a complex number, and $v : \mathbb{R}^N \to \mathbb{R}$ is a suitably chosen penalization function. As explained, we want to penalize mainly the high-frequency components of f; thus, we choose v such that $v(\omega)$ is close to zero for large $||\omega||$.

Hofmann et al. (2008) show that it is possible to select a kernel function κ so that (6) holds, for any function v that satisfies the regularity conditions $v(\omega) \geq 0$, $\int_{\mathbb{R}^N} v(\omega)\, d\omega = 1$, and $v(\omega)$ is symmetric in ω. Specifically, it can be shown that the kernel function

$$\kappa(x_s, x_t) = (2\pi)^{N/2} \tilde{v}(x_s - x_t) \tag{7}$$

satisfies the Mercer (1909) conditions and leads to a norm $||\cdot||_{\mathcal{H}}$ that penalizes non-smoothness as in (6).

We will now discuss two kernels that can be derived using (7). A popular choice is to use

$$v(\omega) = \left(\frac{2\pi}{\sigma^2}\right)^{-N/2} \exp\left(-\frac{\sigma^2}{2}\omega'\omega\right), \tag{8}$$

where $\sigma > 0$ is a tuning parameter. Components of f with frequency ω are penalized more heavily if $||\omega||$ is larger, and high-frequency components are more severely penalized for larger values of σ. It can be shown that substituting (8) into (7) yields

$$\kappa(x_s, x_t) = \exp\left(\frac{-1}{2\sigma^2} ||x_s - x_t||^2\right), \tag{9}$$

where $||\cdot||$ is the usual Euclidean norm. Function (9), introduced by Broomhead and Lowe (1988), is known as the Gaussian kernel.

Notice that the Gaussian kernel allows all frequencies to be present in the prediction function f, albeit with very large penalties for high frequencies. One may alternatively choose to set an infinitely large penalty on certain frequencies ω by setting $v(\omega) = 0$, thereby explicitly disallowing noisy behavior of f.[2] One obvious way to accomplish this is by using the uniform penalty function

$$v(\omega) = \begin{cases} \left(\frac{\sigma}{2}\right)^N & \text{if } -\frac{1}{\sigma} < \omega_n < \frac{1}{\sigma} \text{ for all } n = 1, 2, \ldots, N; \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where again, $\sigma > 0$ is a tuning parameter. Substituting (10) into (7) leads to an expression for the corresponding kernel function,

$$\kappa(x_s, x_t) = \prod_{n=1}^{N} \text{sinc}\left(\frac{x_{sn} - x_{tn}}{\sigma}\right), \tag{11}$$

the Sinc kernel (see Yao, 1967), where $\text{sinc}(0) = 1$ and $\text{sinc}(u) = \sin(u)/u$ for all $u \neq 0$. Despite its intuitive interpretation given in (10), the Sinc kernel does not seem to have found wide application in the kernel literature.

As mentioned before, all kernels discussed in this study have the property that there exists a mapping φ such that $\kappa(x_s, x_t) = \varphi(x_s)' \varphi(x_t)$. However, the kernel functions derived here are much more easily understood by studying how v penalizes certain frequencies than by explicitly finding the regressors $\varphi(x)$. For the sake of completeness, Exterkate et al. (2011) derive the following expression for $\varphi(x)$ for the Gaussian kernel: it contains, for each combination of nonnegative degrees $d_1, d_2, \ldots, d_N$, the "dampened polynomial"

$$\left(\sigma^{-\sum_{n=1}^{N} d_n}\right) \exp\left(\frac{-x'x}{2\sigma^2}\right) \prod_{n=1}^{N} \frac{x_n^{d_n}}{\sqrt{d_n!}}.$$

Finally, one may expand the expression (11) for the Sinc kernel by the classical infinite product $\text{sinc}(u) = \prod_{k=1}^{\infty} \cos\left(u/2^k\right)$. Some algebra using the formula $\cos(u - v) = \cos u \cos v + \sin u \sin v$ then shows that the corresponding $\varphi(x)$ contains all products of the form

$$\prod_{n=1}^{N} \prod_{k=1}^{\infty} f_{nk}\left(\frac{x_n}{2^k}\right), \qquad f_{nk} \in \{\sin, \cos\}.$$

[2] More formally, we avoid division by zero by excluding the region where $v(\omega) = 0$ from the domain of integration in (6), and we restrict f to have $\tilde{f}(\omega) = 0$ in that region.

2.4 Tuning parameters

Two tuning parameters have been introduced in our discussion of kernel ridge regression: a penalization parameter λ and a kernel-specific tuning parameter σ. In this section, we give an interpretation to both of these parameters. This interpretation will result in a small grid of "reasonable" values for both tuning parameters. Selection from this grid, as well as selection of the kernel function, can then be performed using leave-one-out cross-validation; see Cawley and Talbot (2008) for a computationally efficient implementation of this procedure. We choose this selection mechanism because of its close resemblance to the task at hand: the out-of-sample forecasting of the value of the dependent variable for one observation.

The parameter λ is most easily understood from the Bayesian point of view. We assumed that, conditional on x_t and the model parameters, y_t is normally distributed with mean $\varphi(x_t)'\gamma$ and variance θ^2. Equivalently, we may decompose y_t into signal and noise components: $y_t = \varphi(x_t)'\gamma + \varepsilon_t$, with $\varepsilon_t \sim \mathcal{N}\left(0, \theta^2\right)$. The entire analysis in Section 2.2 was conditional on x_t. If we now treat x_t as a random variable, of which the priors on γ and θ are independent, we can write

$$\begin{aligned}\operatorname{var}\left(\varphi(x_t)'\gamma\right) &= \mathrm{E}\left(\varphi(x_t)'\gamma\gamma'\varphi(x_t)\right) = \mathrm{E}\left(\mathrm{E}\left(\varphi(x_t)'\gamma\gamma'\varphi(x_t)\,|x_t\right)\right) \\ &= \mathrm{E}\left(\varphi(x_t)'\left(\frac{\theta^2}{\lambda}I\right)\varphi(x_t)\right) = \frac{\theta^2}{\lambda}\mathrm{E}\left(\varphi(x_t)'\varphi(x_t)\right) = \frac{\theta^2}{\lambda}\mathrm{E}\left(\kappa(x_t, x_t)\right).\end{aligned}$$

This result enables us to relate λ to the signal-to-noise ratio,

$$\psi = \frac{\operatorname{var}\left(\varphi(x_t)'\gamma\right)}{\operatorname{var}(\varepsilon_t)} = \frac{\left(\theta^2/\lambda\right)\mathrm{E}\left(\kappa(x_t, x_t)\right)}{\theta^2} = \frac{\mathrm{E}\left(\kappa(x_t, x_t)\right)}{\lambda}. \tag{12}$$

For the Gaussian kernel (9) and the Sinc kernel (11), $\kappa(x_t, x_t) = 1$ does not depend on x_t and the signal-to-noise ratio is simply $\psi = 1/\lambda$. For the polynomial kernels (5), ψ is also inversely proportional to λ, but the proportionality constant depends on the distribution of x_t. For example, if we assume that $x_t \sim \mathcal{N}(0, I)$, then $x_t'x_t$ follows a χ^2 distribution with N degrees of freedom, and hence

$$\psi = \frac{1}{\lambda}\mathrm{E}\left(\left(1 + \frac{x_t'x_t}{\sigma^2}\right)^d\right) = \frac{1}{\lambda}\sum_{j=0}^{d}\binom{d}{j}\sigma^{-2j}\mathrm{E}\left(\left(x_t'x_t\right)^j\right) = \frac{1}{\lambda}\sum_{j=0}^{d}\binom{d}{j}\sigma^{-2j}\prod_{i=0}^{j-1}(N+2i). \tag{13}$$

We see that in all cases, the "correct" value of λ could be obtained if the signal-to-noise ratio ψ were known. We propose the following simple procedure for estimating ψ: obtain the R^2 from linear

OLS regression of y on a constant and X (or, if N is not small relative to T, on a small number of principal components of X). If the estimated linear model were the true model, we would have $\psi_0 = R^2/\left(1-R^2\right)$, and its corresponding λ_0 can be found using (12). As one expects to obtain a better fit using nonlinear models, it is likely that a $\lambda < \lambda_0$ is required, and we propose to select λ from the grid $\left\{\frac{1}{8}\lambda_0, \frac{1}{4}\lambda_0, \frac{1}{2}\lambda_0, \lambda_0, 2\lambda_0\right\}$. The simulation study in Section 3 confirms that this grid is sufficiently fine.

On the other hand, the parameter σ is best understood in the context of function approximation. For the Gaussian and Sinc kernels, its interpretation is clear from the penalty functions v introduced in the previous section: a higher value of σ forces the prediction function to be smoother. For the Sinc kernel, this works by explicitly disallowing components of frequency greater than $1/\sigma$. Recall that a component of $f(x)$ with a frequency of $1/\sigma$ oscillates $1/(2\pi\sigma)$ times as x changes by one unit. As we will always studentize the predictors, a one-unit change is equivalent to a one-standard-deviation change. We select a grid that implies that such a change in x may never result in more than two oscillations: $\left\{\frac{1}{4\pi}, \frac{1}{2\pi}, \frac{1}{\pi}, \frac{2}{\pi}, \frac{4}{\pi}\right\}$. Again, the Monte Carlo study in Section 3 shows that this five-point grid suffices.

For the Gaussian kernel, although all frequencies are allowed, the penalty function (8) decreases to zero faster for larger values of σ. In fact, 95% of its mass lies in the ball with radius $\sqrt{c_N}/\sigma$ centered at the origin, where c_N is the 95% quantile of the χ^2 distribution with N degrees of freedom. This leaves very little mass (that is, very high penalties) for higher frequencies. Therefore, the same reasoning as above would lead to a grid in which all values are $\sqrt{c_N}$ times those in the grid for the Sinc kernel. As preliminary simulation evidence has indicated that the frequencies outside this 95% ball still have a nonnegligible impact on the estimated prediction function, we shift the grid by one power of two: $\left\{\frac{1}{2}\sqrt{c_N}/\pi, \sqrt{c_N}/\pi, 2\sqrt{c_N}/\pi, 4\sqrt{c_N}/\pi, 8\sqrt{c_N}/\pi\right\}$.

Finally, for the polynomial kernels, the contributions of terms of different orders to the variance of y_t are given in (13). Irrespective of the distribution of x_t, a higher value of σ allows higher-order terms to contribute less. Thus, as for the other kernels, a higher σ imposes more smoothness on the function f. To derive a rule of thumb for the σ grid, we propose that in most applications the first-order effects should dominate in terms of variance contributions, followed by the second-order, third-order, etc. If we assume that $x_t \sim \mathcal{N}(0, I)$, one can derive from the right-hand-side of (13) that this ordering is preserved if $\sigma > \sigma_0 = \sqrt{(d-1)(N+2)/2}$. Thus, for $d > 1$ we select σ from the grid $\left\{\frac{1}{2}\sigma_0, \sigma_0, 2\sigma_0, 4\sigma_0, 8\sigma_0\right\}$. For $d = 1$, this formula yields $\sigma_0 = 0$, which cannot be used. We set $\sigma_0 = \sqrt{N/2}$ instead and construct the grid in the same manner.

3 Monte Carlo simulation

In order to assess the empirical validity of the rules of thumb for selecting tuning parameters described in Section 2.4, and to investigate the impact of kernel choice on forecast quality, we perform two simulation studies. In the first Monte Carlo study, the data-generating processes correspond to the kernels discussed in Section 2.3. For estimation, we consider four different cases:

- treating the kernel and the tuning parameters as known;
- treating the kernel as known, but selecting the tuning parameters using cross-validation;
- using an incorrect kernel, and selecting the tuning parameters using cross-validation; and
- selecting the kernel and the tuning parameters jointly using cross-validation.

In the second Monte Carlo experiment, the data-generating process is such that all kernels estimate a misspecified model. This experiment is intended to resemble practical situations, in which nothing is known about the data-generating process.

3.1 Setup

In each replication of the kernel simulation study, we obtain $T+1$ random draws x_t from the N-variate normal distribution with mean zero and variance the identity matrix. The prediction function $f(x)$ is then defined using the kernel expansion given below equation (1), using random draws $\alpha_t \sim \mathcal{N}(0,1)$ for the expansion coefficients. An additional set of $T+1$ random draws ε_t from the univariate normal distribution is generated, with mean zero and a variance selected to control the signal-to-noise ratio, and $y_t = f(x_t) + \varepsilon_t$ is computed for $t = 1, 2, \ldots, T+1$. Finally, the y_t are rescaled to have mean zero and unit variance. Kernel ridge regression is then used to forecast y_{T+1}, given x_{T+1} and the pairs (y_t, x_t) for $t = 1, 2, \ldots, T$, using the forecast equation (3).

We simulate univariate ($N = 1$), intermediate ($N = 10$), and data-rich ($N = 100$) models, fixing the number of observations at $T = 100$. The kernels that we consider are the polynomial kernels (5) of degrees 1, 2, and 3, the Gaussian kernel (9), and the Sinc kernel (11). The signal-to-noise ratio ψ is varied over $\{0.5, 1, 2\}$, and the smoothness parameter σ is varied over the middle three values in the grids in Section 2.4.

Each kernel is used for forecasting in each data-generating process, to allow us to assess the impact on forecast accuracy of selecting an incorrect kernel. The tuning parameter σ is selected from the grids that we defined in Section 2.4. As the correct value of λ is known in this simulation study, we do not estimate it as described in Section 2.4. Instead, we select it from the grid $\left\{\frac{1}{4}\lambda_0, \frac{1}{2}\lambda_0, \lambda_0, 2\lambda_0, 4\lambda_0\right\}$, where λ_0 is the true value. This procedure allows us to determine whether such a grid, which is of the same form as the grid we proposed for situations in which λ_0 is unknown, is sufficiently fine.

In the second simulation study, we consider the univariate model $y_t = \left(1 + \exp\left(-10x_t\right)\right)^{-1} + \varepsilon_t$. We shall refer to this experiment as the logistic simulation study. The factor 10 in the exponent is present to make the data-generating process sufficiently nonlinear, see also Figure 1. Note that in this case, the true model differs substantially from the prediction functions associated with each of the kernels. As $|x|$ grows large, a prediction function estimated using a polynomial kernel has $|f(x)| \to \infty$, while the Gaussian and Sinc kernels both have $f(x) \to 0$. In contrast, the logistic function approaches two different finite values: $f(x) \to 0$ as $x \to -\infty$ and $f(x) \to 1$ as $x \to \infty$.

As in the kernel simulation study, we vary the signal-to-noise ratio ψ over $\{0.5, 1, 2\}$, and we set $T = 100$. Forecasts are obtained using the same five kernels as above, selecting the tuning parameters λ and σ as described in Section 2.4.

3.2 Results

Mean squared prediction errors (MSPEs) over 5000 replications of the kernel simulation study are shown in Tables A.1-A.3 in Appendix A, and a summary of these results is reported in Table 1. For ease of comparison, we have divided all MSPEs by $1/(\psi + 1)$, the expected MSPE that would result if the data-generating process were known and used. The summarized results in Table 1 were obtained by averaging the relative MSPEs over all data-generating processes (DGPs) with the same kernel and number of predictors; the differences across different values of the parameters ψ and σ are minor.

The rows labeled "kernel, λ, σ correct" list the MSPEs that are obtained if kernel ridge regression is used with the same kernel and tuning parameter σ as in the DGP, and with the value of λ corresponding to the true signal-to-noise ratio. As we would expect by our normalization, most numbers in these rows are close to unity.

We now shift attention to the MSPEs resulting from using the correct kernel, but selecting λ and σ using cross-validation, which are indicated in boldface in the rows labeled "selecting λ and σ using

Table 1: Average relative mean squared prediction errors in the kernel simulation study.

		DGP: Poly(1)			DGP: Poly(2)			DGP: Poly(3)			DGP: Gauss			DGP: Sinc		
	$N =$	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100
kernel, λ, σ correct		0.99	1.10	1.07	1.00	1.14	1.05	1.01	1.12	1.03	1.02	1.11	1.01	1.04	1.02	0.97
	Poly(1)	**0.99**	**1.08**	**1.04**	1.10	1.12	1.05	1.08	1.10	1.04	1.20	1.16	1.03	1.60	1.08	1.01
selecting	Poly(2)	1.00	1.10	1.04	**1.00**	**1.12**	**1.04**	1.01	1.11	1.03	1.14	1.20	1.04	1.51	1.10	1.02
λ and σ	Poly(3)	1.01	1.10	1.04	1.02	1.12	1.04	**1.02**	**1.11**	**1.03**	1.11	1.20	1.03	1.47	1.10	1.01
using CV	Gauss	1.02	1.10	1.03	1.04	1.12	1.03	1.05	1.11	1.02	**1.03**	**1.13**	**1.02**	1.06	1.08	1.00
	Sinc	1.03	1.29	2.07	1.05	1.29	1.89	1.06	1.28	1.91	1.04	1.20	1.54	**1.06**	**1.06**	**0.97**
selecting	poly	**1.01**	**1.10**	**1.04**	**1.01**	**1.13**	**1.05**	**1.02**	**1.12**	**1.04**	1.11	1.21	1.04	1.47	1.10	1.02
kernel, λ, σ	non-poly	1.02	1.10	1.03	1.05	1.13	1.03	1.05	1.11	1.02	**1.04**	**1.14**	**1.02**	**1.07**	**1.07**	**1.00**
using CV	all	1.02	1.10	1.04	1.03	1.13	1.05	1.03	1.12	1.04	1.04	1.15	1.03	1.07	1.08	1.01

Notes: This table reports mean squared prediction errors (MSPEs) over 5000 replications of the kernel simulation study, relative to the expected value of the MSPE if the DGP would be known, which is $1/(\psi + 1)$. The MSPEs in this table were obtained by averaging over all values of the DGP parameters σ and ψ; detailed tables are shown in Appendix A. MSPEs obtained using the correct kernel (in the group of rows labeled "selecting λ and σ using CV") or the correct type of kernel ("selecting kernel, λ, σ using CV") are printed in boldface.

CV" (where CV stands for cross-validation). Interestingly, these numbers are not much different from those obtained when fixing λ and σ at their correct values; we find that not knowing the correct values of these parameters leads to an increase in MSPE of only around 0.4%. Recall that the values of λ and σ are selected from a grid that allows each of them to be off by a factor of four. Thus, while very extreme values of the tuning parameters might lead to poor forecasts, our relatively crude rule of thumb for selecting their values seems sufficient. Inspecting the selected values, we find that λ is generally selected equal to or somewhat larger than the value corresponding to the data-generating process, whereas for σ a larger value than that in the DGP is often selected in all kernels.[3] This suggests that kernel ridge regression is somewhat biased toward smoother prediction functions, although the effect of this bias on forecast accuracy is minor.

Next, we investigate what happens if we use an incorrect kernel. The results from this procedure can be found in the group of rows labeled "selecting λ and σ using CV", excluding the numbers printed in boldface. Four features clearly emerge from these results. First, we observe that if the data-generating process is polynomial, using a polynomial kernel of too high degree hardly hurts the forecasting performance. Apparently, the ridge term is an effective safeguard against overfitting in this case. Using a polynomial kernel of too low degree does deteriorate the quality of the forecasts, as expected, especially in the one-predictor case. Second, although the Gaussian and Sinc kernels estimate quite similar

[3]Tables listing these selection frequencies are not included for brevity. They may be obtained from the author upon request.

"smooth" prediction functions, the performance of the Gaussian kernel is often much better. Third, there is an important difference between polynomial and non-polynomial kernels. Using a kernel from one group when the data is generated from a process in the other group almost invariably leads to large forecast errors. Fourth, we observe from the full tables in Appendix A that the differences between kernels are mitigated if the true value of σ goes up. Notice that for all types of kernels under consideration, a higher value of σ translates into a smoother prediction function. The smoother a function is, the less the estimation method matters.

In the last rows of Table 1, labeled "selecting kernel, λ, σ using CV", we show the results from selecting not only the tuning parameters, but also the kernel function using cross-validation. We find that when selecting among all five kernels, the cross-validation procedure selects the correct kernel in about 40% of the cases. Moreover, incorrect choices usually fall in the correct group of polynomials or non-polynomials. As a result, the MSPEs (shown in the last row of the table) are on average less than 1.5% larger than when the correct kernel is imposed. The selection frequency of the correct kernel is lower for larger values of σ; again, the smoothest functions are easily estimable using any method.

We finally consider the differences that arise if we vary the set of kernels among which cross-validation selects. If only the correct group of kernels (that is, polynomial or non-polynomial; the boldface numbers in the rows labeled "poly" or "non-poly" in Table 1) is available for cross-validation, the correct selection frequency rises to about 60%, and the MSPE increase compared to using the correct kernel is under 1%. However, choosing the wrong group can make a difference: estimating a Gaussian or Sinc prediction function if the DGP is polynomial has only a small influence on the MSPEs, but the reverse case leads to much larger increases. For this reason, and because of their associated smooth and flexible functional forms, we argue that a practitioner is in general better off using only the Gaussian and Sinc kernels, unless he would have strong prior knowledge that the true predictive relation is polynomial.

We now turn to the results of the logistic simulation study, in which kernel ridge regression always estimates an incorrectly specified model. The relative MSPEs, again over 5000 replications, are reported in Table 2. It is clear from theses results that the Gaussian kernel approximates the logistic function best, with the Sinc kernel ranking second best. For the smallest signal-to-noise ratio that we consider ($\psi = 0.5$), the differences between the kernels are minor. As ψ increases, however, the polynomial kernels perform much worse than the non-polynomial ones. If the DGP is reflected by the data more clearly, it becomes more apparent that a polynomial prediction function is not a suitable approximation.

Table 2: Relative mean squared prediction errors in the logistic simulation study.

		$\psi = 0.5$	$\psi = 1.0$	$\psi = 2.0$
	Poly(1)	1.06	1.12	1.25
selecting	Poly(2)	1.07	1.13	1.25
λ and σ	Poly(3)	1.07	1.11	1.19
using CV	Gauss	1.05	1.07	1.09
	Sinc	1.06	1.09	1.12
selecting	poly	1.07	1.11	1.19
kernel, λ, σ	non-poly	1.06	1.08	1.10
using CV	all	1.06	1.09	1.10

Notes: This table reports mean squared prediction errors (MSPEs) over 5000 replications of the logistic simulation study, relative to the expected value of the MSPE if the DGP would be known, which is $1/(\psi + 1)$.

Selecting the kernel using cross-validation leads to a forecast accuracy that is almost as good as imposing the Gaussian kernel, as long as this kernel is in the set of competing models. Unsurprisingly, cross-validation selects the Gaussian kernel in 58% and the Sinc kernel in 33% of the replications. The remaining 9% of the cases, in which polynomial kernels are selected, still has a slight negative impact on the forecast accuracy. This result illustrates our recommendation that in general, superior results can be obtained if one looks beyond the popular polynomial kernels.

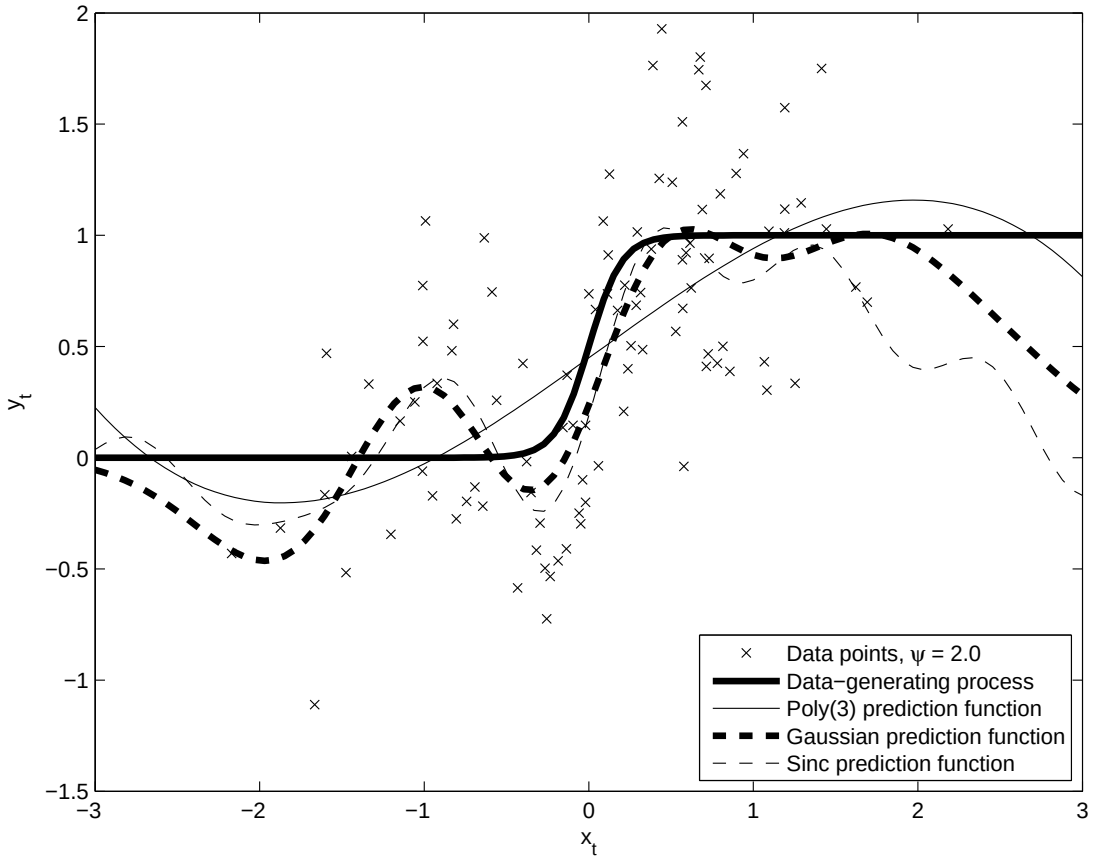
As an illustrative example, we show a scatter plot of one simulated data set in Figure 1. The true prediction function f is also shown, as well as its estimates using the third-degree polynomial, Gaussian, and Sinc kernels. This figure shows that in contrast with the non-polynomial estimates, the polynomial prediction function is not sufficiently flexible to capture the behavior of the true f. This is particularly evident near $x_t = 0$, where most data points are located.

4 Conclusion

We review the technique of kernel ridge regression from two different points of view, namely from a function approximation perspective and from a Bayesian standpoint. This combination of perspectives enables us to give a clear interpretation to two tuning parameters that are generally present in kernel ridge regression. We relate one of these parameters to the signal-to-noise ratio, and the other to the smoothness of the regression function. Moreover, we provide rules of thumb for selecting their values.

In addition to the well-known polynomial and Gaussian kernels, we discuss the Sinc kernel. Kernel ridge regression using this kernel function acts as a low-pass filter, so that any high-frequency patterns

Figure 1: The logistic data-generating process with 100 data points, generated with signal-to-noise ratio $\psi = 2.0$. Three estimated prediction functions, using the Poly(3), Gaussian, and Sinc kernels, are also shown.



observed in the data are considered noise and are discarded. Despite this attractive feature, the Sinc kernel has not received widespread attention in the kernel literature.

Our simulation studies confirm the empirical usefulness of our parameter selection rules. Compared to using the true values of the tuning parameters, selecting their values using our rules of thumb leads to an increase of mean squared prediction errors of only 0.4%.

Cross-validation can also be used relatively safely to distinguish among different kernel functions, with a 1.5% increase in mean squared prediction errors when compared to using the correct kernel. We argue that this problem is in large part due to the large difference between non-polynomial and polynomial kernels; if it is known in which of these classes to search, the MSPE increase is limited to less than 1%. For this reason, and because of their smoother and more flexible prediction functions, we recommend the use of non-polynomial kernels in practice, if no prior knowledge of the true prediction function is available.

A Detailed simulation results

In this appendix we report the mean squared prediction errors for all data-generating processes in the kernel simulation study. A summary of these results was presented in Table 1.

Table A.1: Relative mean squared prediction errors in the kernel simulation study, for $T = 100, N = 1$.

		DGP: Poly(1)									DGP: Poly(2)								
		$\sigma = 0.71$			$\sigma = 1.41$			$\sigma = 2.83$			$\sigma = 1.22$			$\sigma = 2.45$			$\sigma = 4.90$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.01	1.01	1.01	1.00	1.00	1.01	0.99	0.99	1.00
	Poly(1)	**0.99**	**0.99**	**0.99**	**1.00**	**1.00**	**0.99**	**0.99**	**0.99**	**0.99**	1.12	1.24	1.49	1.01	1.02	1.04	0.99	0.99	0.99
selecting	Poly(2)	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.00**	**1.00**	**1.00**
λ and σ	Poly(3)	1.02	1.02	1.02	1.01	1.01	1.01	1.01	1.01	1.01	1.02	1.02	1.03	1.01	1.02	1.02	1.01	1.01	1.01
using CV	Gauss	1.02	1.03	1.04	1.01	1.02	1.02	1.01	1.01	1.02	1.06	1.09	1.12	1.02	1.02	1.03	1.01	1.01	1.01
	Sinc	1.02	1.03	1.04	1.02	1.03	1.04	1.02	1.02	1.03	1.07	1.10	1.14	1.03	1.03	1.04	1.02	1.02	1.03
selecting	poly	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.02**	**1.02**	**1.02**	**1.01**	**1.02**	**1.02**	**1.01**	**1.01**	**1.01**
kernel, λ, σ	non-poly	1.02	1.03	1.04	1.02	1.02	1.03	1.01	1.02	1.02	1.06	1.09	1.12	1.02	1.03	1.03	1.01	1.02	1.02
using CV	all	1.02	1.02	1.03	1.02	1.02	1.03	1.02	1.02	1.02	1.04	1.04	1.06	1.03	1.03	1.03	1.02	1.02	1.02

		DGP: Poly(3)									DGP: Gauss								
		$\sigma = 1.73$			$\sigma = 3.46$			$\sigma = 6.93$			$\sigma = 0.62$			$\sigma = 1.25$			$\sigma = 2.50$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		1.02	1.02	1.03	1.00	1.00	1.01	0.99	0.99	0.99	1.02	1.03	1.04	1.01	1.01	1.02	0.99	1.00	1.00
	Poly(1)	1.10	1.21	1.42	1.01	1.02	1.03	0.99	0.99	0.99	1.18	1.37	1.73	1.07	1.13	1.26	1.01	1.02	1.04
selecting	Poly(2)	1.02	1.04	1.06	1.00	1.00	1.01	1.00	1.00	1.00	1.15	1.28	1.54	1.04	1.07	1.13	1.01	1.01	1.02
λ and σ	Poly(3)	**1.02**	**1.03**	**1.03**	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	**1.01**	1.12	1.22	1.41	1.03	1.05	1.07	1.01	1.02	1.03
using CV	Gauss	1.07	1.10	1.15	1.02	1.02	1.03	1.01	1.01	1.01	**1.04**	**1.05**	**1.05**	**1.02**	**1.03**	**1.03**	**1.01**	**1.02**	**1.02**
	Sinc	1.08	1.11	1.16	1.02	1.03	1.04	1.02	1.02	1.02	1.05	1.06	1.07	1.03	1.03	1.04	1.02	1.02	1.03
selecting	poly	**1.03**	**1.03**	**1.03**	**1.01**	**1.01**	**1.02**	**1.01**	**1.01**	**1.01**	1.12	1.22	1.41	1.04	1.05	1.07	1.02	1.02	1.03
kernel, λ, σ	non-poly	1.07	1.10	1.15	1.02	1.03	1.03	1.01	1.01	1.02	**1.05**	**1.06**	**1.07**	**1.03**	**1.03**	**1.04**	**1.02**	**1.02**	**1.03**
using CV	all	1.05	1.05	1.06	1.02	1.03	1.03	1.01	1.02	1.02	1.05	1.06	1.07	1.03	1.04	1.04	1.02	1.03	1.03

		DGP: Sinc								
		$\sigma = 0.16$			$\sigma = 0.32$			$\sigma = 0.64$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		1.05	1.07	1.08	1.02	1.03	1.04	1.01	1.01	1.02
	Poly(1)	1.38	1.76	2.52	1.28	1.56	2.13	1.11	1.21	1.42
selecting	Poly(2)	1.37	1.73	2.45	1.25	1.48	1.95	1.06	1.11	1.20
λ and σ	Poly(3)	1.37	1.72	2.42	1.23	1.44	1.85	1.05	1.06	1.09
using CV	Gauss	1.07	1.09	1.10	1.04	1.05	1.07	1.03	1.03	1.04
	Sinc	**1.08**	**1.09**	**1.11**	**1.06**	**1.06**	**1.07**	**1.03**	**1.04**	**1.04**
selecting	poly	1.38	1.72	2.42	1.24	1.44	1.86	1.05	1.06	1.09
kernel, λ, σ	non-poly	**1.08**	**1.09**	**1.11**	**1.05**	**1.06**	**1.08**	**1.03**	**1.04**	**1.04**
using CV	all	1.08	1.09	1.11	1.05	1.06	1.07	1.04	1.04	1.05

Notes: This table reports mean squared prediction errors (MSPEs) over 5000 replications of the kernel simulation study with $N = 1$ predictor, relative to the expected value of the MSPE if the DGP would be known, which is $1/(\psi + 1)$. MSPEs obtained using the correct kernel (in the group of rows labeled "selecting λ and σ using CV") or the correct type of kernel ("selecting kernel, λ, σ using CV") are printed in boldface.

Table A.2: Relative mean squared prediction errors in the kernel simulation study, for $T = 100, N = 10$.

		DGP: Poly(1)									DGP: Poly(2)								
		$\sigma = 2.24$			$\sigma = 4.47$			$\sigma = 8.94$			$\sigma = 2.45$			$\sigma = 4.90$			$\sigma = 9.80$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		1.11	1.12	1.14	1.08	1.10	1.12	1.05	1.07	1.09	1.13	1.20	1.31	1.09	1.12	1.15	1.06	1.08	1.10
	Poly(1)	**1.11**	**1.12**	**1.13**	**1.07**	**1.08**	**1.09**	**1.05**	**1.05**	**1.06**	1.13	1.18	1.27	1.08	1.09	1.11	1.05	1.06	1.07
selecting	Poly(2)	1.12	1.15	1.15	1.08	1.09	1.10	1.05	1.06	1.07	**1.15**	**1.19**	**1.25**	**1.09**	**1.11**	**1.12**	**1.06**	**1.07**	**1.08**
λ and σ	Poly(3)	1.12	1.14	1.16	1.08	1.09	1.11	1.06	1.06	1.06	1.15	1.19	1.25	1.09	1.11	1.12	1.06	1.07	1.07
using CV	Gauss	1.12	1.14	1.16	1.08	1.09	1.11	1.06	1.06	1.07	1.14	1.19	1.26	1.09	1.11	1.12	1.06	1.07	1.08
	Sinc	1.19	1.30	1.48	1.17	1.26	1.42	1.16	1.25	1.41	1.19	1.29	1.46	1.17	1.27	1.43	1.16	1.25	1.41
selecting	poly	**1.13**	**1.14**	**1.15**	**1.08**	**1.09**	**1.11**	**1.06**	**1.06**	**1.07**	**1.16**	**1.20**	**1.26**	**1.09**	**1.11**	**1.13**	**1.07**	**1.07**	**1.08**
kernel, λ, σ	non-poly	1.12	1.14	1.16	1.08	1.09	1.11	1.06	1.06	1.07	1.14	1.19	1.26	1.09	1.11	1.12	1.06	1.07	1.08
using CV	all	1.13	1.15	1.15	1.09	1.10	1.11	1.06	1.07	1.07	1.16	1.20	1.26	1.10	1.11	1.13	1.07	1.08	1.08

		DGP: Poly(3)									DGP: Gauss								
		$\sigma = 3.46$			$\sigma = 6.93$			$\sigma = 13.86$			$\sigma = 1.36$			$\sigma = 2.72$			$\sigma = 5.45$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		1.12	1.17	1.25	1.08	1.11	1.14	1.05	1.07	1.09	1.07	1.09	1.11	1.10	1.14	1.19	1.07	1.09	1.12
	Poly(1)	1.12	1.16	1.24	1.07	1.08	1.10	1.05	1.05	1.06	1.13	1.21	1.37	1.11	1.16	1.25	1.06	1.07	1.08
selecting	Poly(2)	1.14	1.18	1.23	1.09	1.10	1.11	1.06	1.06	1.07	1.17	1.28	1.51	1.13	1.18	1.27	1.07	1.08	1.09
λ and σ	Poly(3)	**1.13**	**1.17**	**1.22**	**1.08**	**1.10**	**1.11**	**1.06**	**1.06**	**1.07**	1.17	1.28	1.51	1.13	1.18	1.27	1.07	1.08	1.10
using CV	Gauss	1.12	1.16	1.22	1.08	1.09	1.11	1.06	1.06	1.07	**1.11**	**1.15**	**1.21**	**1.12**	**1.15**	**1.19**	**1.07**	**1.08**	**1.10**
	Sinc	1.17	1.26	1.42	1.17	1.26	1.43	1.16	1.25	1.41	1.12	1.18	1.28	1.13	1.17	1.22	1.14	1.21	1.33
selecting	poly	**1.15**	**1.19**	**1.24**	**1.09**	**1.10**	**1.12**	**1.06**	**1.07**	**1.07**	1.18	1.29	1.53	1.13	1.19	1.28	1.07	1.08	1.10
kernel, λ, σ	non-poly	1.12	1.16	1.22	1.08	1.09	1.11	1.06	1.06	1.07	**1.12**	**1.16**	**1.22**	**1.12**	**1.16**	**1.20**	**1.07**	**1.09**	**1.10**
using CV	all	1.15	1.19	1.24	1.09	1.10	1.12	1.07	1.07	1.07	1.13	1.17	1.23	1.13	1.16	1.21	1.08	1.09	1.11

		DGP: Sinc								
		$\sigma = 0.16$			$\sigma = 0.32$			$\sigma = 0.64$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		1.01	1.01	1.01	1.01	1.01	1.01	1.03	1.04	1.04
	Poly(1)	1.04	1.06	1.08	1.05	1.06	1.08	1.07	1.11	1.17
selecting	Poly(2)	1.06	1.08	1.10	1.06	1.08	1.10	1.09	1.14	1.22
λ and σ	Poly(3)	1.06	1.07	1.10	1.06	1.07	1.10	1.09	1.14	1.23
using CV	Gauss	1.05	1.06	1.09	1.05	1.06	1.09	1.07	1.10	1.15
	Sinc	**1.03**	**1.04**	**1.06**	**1.03**	**1.04**	**1.06**	**1.06**	**1.08**	**1.11**
selecting	poly	1.06	1.08	1.10	1.06	1.08	1.10	1.09	1.14	1.23
kernel, λ, σ	non-poly	**1.05**	**1.06**	**1.09**	**1.05**	**1.06**	**1.09**	**1.07**	**1.08**	**1.12**
using CV	all	1.05	1.07	1.09	1.06	1.07	1.09	1.07	1.09	1.12

Notes: This table reports relative MSPEs over 5000 replications of the kernel simulation study with $N = 10$ predictors.

Table A.3: Relative mean squared prediction errors in the kernel simulation study, for $T = 100, N = 100$.

		DGP: Poly(1)									DGP: Poly(2)								
		$\sigma = 7.07$			$\sigma = 14.14$			$\sigma = 28.28$			$\sigma = 7.14$			$\sigma = 14.28$			$\sigma = 28.57$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		1.06	1.13	1.25	1.01	1.05	1.12	0.98	0.99	1.02	1.02	1.05	1.08	1.02	1.06	1.13	0.99	1.01	1.05
	Poly(1)	**1.05**	**1.09**	**1.16**	**1.01**	**1.03**	**1.05**	**1.00**	**1.00**	**1.01**	1.04	1.08	1.16	1.02	1.04	1.08	1.00	1.01	1.02
selecting	Poly(2)	1.05	1.08	1.14	1.02	1.03	1.05	1.00	1.00	1.01	**1.03**	**1.06**	**1.10**	**1.02**	**1.04**	**1.07**	**1.00**	**1.01**	**1.02**
λ and σ	Poly(3)	1.04	1.08	1.14	1.01	1.03	1.05	1.00	1.00	1.01	1.02	1.05	1.08	1.02	1.04	1.07	1.00	1.01	1.02
using CV	Gauss	1.03	1.07	1.13	1.01	1.02	1.04	1.00	1.00	1.01	1.02	1.04	1.07	1.01	1.03	1.06	1.00	1.01	1.02
	Sinc	1.40	1.83	2.69	1.46	1.94	2.91	1.47	1.97	2.97	1.21	1.46	1.95	1.42	1.87	2.77	1.47	1.96	2.94
selecting	poly	**1.05**	**1.09**	**1.16**	**1.02**	**1.03**	**1.05**	**1.00**	**1.00**	**1.01**	**1.04**	**1.08**	**1.15**	**1.02**	**1.04**	**1.08**	**1.01**	**1.01**	**1.02**
kernel, λ, σ	non-poly	1.03	1.07	1.13	1.01	1.02	1.04	1.00	1.00	1.01	1.02	1.04	1.07	1.01	1.03	1.06	1.00	1.01	1.02
using CV	all	1.05	1.09	1.16	1.02	1.03	1.05	1.00	1.00	1.01	1.04	1.08	1.15	1.02	1.05	1.08	1.01	1.01	1.02

		DGP: Poly(3)									DGP: Gauss								
		$\sigma = 10.10$			$\sigma = 20.20$			$\sigma = 40.40$			$\sigma = 3.55$			$\sigma = 7.10$			$\sigma = 14.20$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		1.01	1.03	1.06	1.01	1.05	1.11	0.99	1.00	1.03	0.97	0.97	0.97	1.00	1.01	1.02	1.00	1.03	1.08
	Poly(1)	1.03	1.06	1.12	1.02	1.03	1.06	1.00	1.01	1.01	0.99	1.00	1.03	1.02	1.05	1.11	1.01	1.02	1.04
selecting	Poly(2)	1.02	1.05	1.08	1.02	1.03	1.06	1.00	1.01	1.02	1.00	1.02	1.05	1.02	1.05	1.10	1.01	1.02	1.04
λ and σ	Poly(3)	**1.02**	**1.04**	**1.07**	**1.01**	**1.03**	**1.06**	**1.00**	**1.01**	**1.02**	1.00	1.01	1.03	1.02	1.04	1.10	1.01	1.02	1.04
using CV	Gauss	1.01	1.02	1.04	1.01	1.03	1.05	1.00	1.01	1.01	**0.99**	**1.00**	**1.02**	**1.01**	**1.04**	**1.08**	**1.01**	**1.02**	**1.05**
	Sinc	1.22	1.47	1.97	1.43	1.89	2.81	1.47	1.96	2.95	0.97	0.97	0.97	1.23	1.50	2.02	1.45	1.93	2.87
selecting	poly	**1.03**	**1.06**	**1.12**	**1.02**	**1.03**	**1.06**	**1.00**	**1.01**	**1.02**	1.00	1.02	1.04	1.03	1.05	1.11	1.01	1.03	1.04
kernel, λ, σ	non-poly	1.01	1.02	1.04	1.01	1.03	1.05	1.00	1.01	1.01	**0.99**	**1.00**	**1.02**	**1.01**	**1.04**	**1.08**	**1.01**	**1.02**	**1.05**
using CV	all	1.03	1.06	1.12	1.02	1.03	1.06	1.00	1.01	1.02	0.99	1.01	1.03	1.02	1.05	1.10	1.01	1.03	1.05

		DGP: Sinc								
		$\sigma = 0.16$			$\sigma = 0.32$			$\sigma = 0.64$		
	$\psi =$	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
kernel, λ, σ correct		0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
	Poly(1)	0.99	1.00	1.03	0.99	1.00	1.03	0.99	1.00	1.03
selecting	Poly(2)	1.00	1.02	1.04	1.00	1.02	1.04	1.00	1.02	1.04
λ and σ	Poly(3)	0.99	1.01	1.03	0.99	1.01	1.03	0.99	1.01	1.03
using CV	Gauss	0.98	1.00	1.01	0.98	1.00	1.01	0.98	1.00	1.01
	Sinc	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**
selecting	poly	1.00	1.02	1.04	1.00	1.02	1.04	1.00	1.02	1.04
kernel, λ, σ	non-poly	**0.98**	**1.00**	**1.01**	**0.98**	**1.00**	**1.01**	**0.98**	**1.00**	**1.01**
using CV	all	0.99	1.01	1.03	0.99	1.01	1.03	0.99	1.01	1.03

Notes: This table reports relative MSPEs over 5000 replications of the kernel simulation study with $N = 100$ predictors.

References

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68: 337–404, 1950.

B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the Annual Conference on Computational Learning Theory*, pages 144–152. ACM Press, Pittsburgh, Pennsylvania, 1992.

D.S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

G.C. Cawley and N.L.C. Talbot. Efficient approximate leave-one-out cross-validation for kernel logistic regression. *Machine Learning*, 71:243–264, 2008.

P. Exterkate, P.J.F. Groenen, C. Heij, and D. van Dijk. Nonlinear forecasting with many predictors using kernel ridge regression. *Tinbergen Institute Discussion Paper TI 11-007/4*, 2011.

T. Hofmann, B. Schölkopf, and A.J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36:1171–1220, 2008.

G.S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

A.B. Kock and T. Teräsvirta. Forecasting with non-linear models. In M.P. Clements and D.F. Hendry, editors, *Oxford Handbook of Economic Forecasting*, pages 61–87. Oxford University Press, Oxford, 2011.

S.C. Ludvigson and S. Ng. The empirical risk-return relation: A factor analysis approach. *Journal of Financial Economics*, 83:171–222, 2007.

S.C. Ludvigson and S. Ng. Macro factors in bond risk premia. *Review of Financial Studies*, 22:5027–5067, 2009.

M.C. Medeiros, T. Teräsvirta, and G. Rech. Building neural network models for time series: A statistical approach. *Journal of Forecasting*, 25:49–75, 2006.

J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London: Series A*, 209:415–446, 1909.

T. Poggio. On optimal nonlinear associative recall. *Biological Cybernetics*, 19:201–209, 1975.

H. Raiffa and R. Schlaifer. *Applied Statistical Decision Theory*. Harvard University Press, Cambridge, Massachusetts, 1961.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

A.J. Smola, B. Schölkopf, and K.-R. Müller. The connection between regularizaton operators and support vector kernels. *Neural Networks*, 11:637–649, 1998.

J.H. Stock and M.W. Watson. A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series. In R.F. Engle and H. White, editors, *Cointegration, Causality and Forecasting. A Festschrift in Honour of Clive W.J. Granger*, pages 1–44. Oxford University Press, Oxford, 1999.

J.H. Stock and M.W. Watson. Macroeconomic forecasting using diffusion indexes. *Journal of Business and Economic Statistics*, 20:147–162, 2002.

T. Teräsvirta. Forecasting economic variables with nonlinear models. In G. Elliot, C.W.J. Granger, and A. Timmermann, editors, *Handbook of Economic Forecasting*, pages 413–458. Elsevier, Amsterdam, 2006.

T. Teräsvirta, D. van Dijk, and M.C. Medeiros. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *International Journal of Forecasting*, 21:755–774, 2005.

H. White. Approximate nonlinear forecasting methods. In G. Elliot, C.W.J. Granger, and A. Timmermann, editors, *Handbook of Economic Forecasting*, pages 459–514. Elsevier, Amsterdam, 2006.

K. Yao. Applications of reproducing kernel Hilbert spaces: Bandlimited signal models. *Information and Control*, 11:429–444, 1967.

Research Papers 2012

CREATES
Center for Research in Econometric Analysis of Time Series

2011-48: Christian M. Dahl, Daniel le Maire and Jakob R. Munch: Wage Dispersion and Decentralization of Wage Bargaining

2011-49: Torben G. Andersen, Oleg Bondarenko and Maria T. Gonzalez-Perez: Coherent Model-Free Implied Volatility: A Corridor Fix for High-Frequency VIX

2011-50: Torben G. Andersen and Oleg Bondarenko: VPIN and the Flash Crash

2011-51: Tim Bollerslev, Daniela Osterrieder, Natalia Sizova and George Tauchen: Risk and Return: Long-Run Relationships, Fractional Cointegration, and Return Predictability

2011-52: Lars Stentoft: What we can learn from pricing 139,879 Individual Stock Options

2011-53: Kim Christensen, Mark Podolskij and Mathias Vetter: On covariation estimation for multivariate continuous Itô semimartingales with noise in non-synchronous observation schemes

2012-01: Matei Demetrescu and Robinson Kruse: The Power of Unit Root Tests Against Nonlinear Local Alternatives

2012-02: Matias D. Cattaneo, Michael Jansson and Whitney K. Newey: Alternative Asymptotics and the Partially Linear Model with Many Regressors

2012-03: Matt P. Dziubinski: Conditionally-Uniform Feasible Grid Search Algorithm

2012-04: Jeroen V.K. Rombouts, Lars Stentoft and Francesco Violante: The Value of Multivariate Model Sophistication: An Application to pricing Dow Jones Industrial Average options

2012-05: Anders Bredahl Kock: On the Oracle Property of the Adaptive LASSO in Stationary and Nonstationary Autoregressions

2012-06: Christian Bach and Matt P. Dziubinski: Commodity derivatives pricing with inventory effects

2012-07: Cristina Amado and Timo Teräsvirta: Modelling Changes in the Unconditional Variance of Long Stock Return Series

2012-08: Anne Opschoor, Michel van der Wel, Dick van Dijk and Nick Taylor: On the Effects of Private Information on Volatility

2012-09: Annastiina Silvennoinen and Timo Teräsvirta: Modelling conditional correlations of asset returns: A smooth transition approach

2012-10: Peter Exterkate: Model Selection in Kernel Ridge Regression