Nonlinear Forecasting With Many Predictors Using Kernel Ridge Regression*

Peter Exterkate [a, †] Patrick J.F. Groenen [b] Christiaan Heij [b] Dick van Dijk [b]

[a] *CREATES, Aarhus University, Denmark*
[b] *Econometric Institute, Erasmus University Rotterdam, The Netherlands*

February 29, 2012

Abstract

This paper puts forward kernel ridge regression as an approach for forecasting with many predictors that are related nonlinearly to the target variable. In kernel ridge regression, the observed predictor variables are mapped nonlinearly into a high-dimensional space, where estimation of the predictive regression model is based on a shrinkage estimator to avoid overfitting. We extend the kernel ridge regression methodology to enable its use for economic time-series forecasting, by including lags of the dependent variable or other individual variables as predictors, as typically desired in macroeconomic and financial applications. Monte Carlo simulations as well as an empirical application to various key measures of real economic activity confirm that kernel ridge regression can produce more accurate forecasts than traditional linear methods for dealing with many predictors based on principal component regression.

Keywords: High dimensionality, nonlinear forecasting, ridge regression, kernel methods.
JEL Classification: C53, C63, E27.

*We thank conference participants at the International Conferences on Computational and Financial Econometrics (Limassol, October 2009, and London, December 2010), at the Eurostat Colloquium on Modern Tools for Business Cycle Analysis (Luxembourg, September 2010), at the International Conference on High-Dimensional Econometric Modelling at Cass Business School (London, December 2010), at the Applied Time Series Econometrics Workshop at the Federal Reserve Bank of St. Louis (St. Louis, April 2011), at the Netherlands Econometric Study Group (Rotterdam, June 2011), at the International Symposium on Forecasting (Prague, June 2011), and at the Info-Metrics Workshop on Information Theory and Shrinkage Estimation (Washington DC, November 2011) for useful comments and suggestions. Exterkate acknowledges support from CREATES, funded by the Danish National Research Foundation.

†Corresponding author. 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 current practice, forecasters in macroeconomics and finance face a trade-off between model complexity and forecast accuracy. Due to the uncertainty associated with model choice and parameter estimation, a highly complex predictive regression model based on many variables or intricate nonlinear structures is often found to produce less accurate forecasts than a simpler model that ignores major parts of the information that is at the researcher's disposal. Various methods for working with many predictors while circumventing this *curse of dimensionality* in a linear framework have been applied in the recent forecasting literature, as surveyed by Stock and Watson (2006). Most prominently, Stock and Watson (2002) advocate summarizing large panels of predictor variables into a small number of principal components, which are then used for forecasting purposes in a dynamic factor model. Alternative approaches include combining forecasts based on multiple models, each including only a small number of variables (Faust and Wright, 2009; Wright, 2009; Aiolfi and Favero, 2005; Huang and Lee, 2010; Rapach et al., 2010), partial least squares (Groen and Kapetanios, 2008), and Bayesian regression (De Mol et al., 2008; Bańbura et al., 2010; Carriero et al., 2011). Stock and Watson (2009) find that for forecasting macroeconomic time series, the dynamic factor model approach is preferable to these alternatives; see also Ludvigson and Ng (2007, 2009) and Çakmaklı and van Dijk (2010) for successful applications in finance.

The possibility of nonlinear relations among macroeconomic and financial time series has also received ample attention during the last two decades. Among the most popular nonlinear forecast methods are regime-switching models and neural networks, see the surveys by Teräsvirta (2006) and White (2006), respectively, and the comprehensive overview by Kock and Teräsvirta (2011). Typically, these approaches are only suitable for a small number of predictors, and their ability to improve upon the predictive accuracy of linear forecasting techniques seems limited, see Stock and Watson (1999), Medeiros et al. (2006), and Teräsvirta et al. (2005), among others. Giovannetti (2011) proposes a hybrid approach, estimating a nonlinear model using principal components extracted from a large set of predictors.

In this paper, we introduce a forecasting technique that can deal with high-dimensionality and nonlinearity simultaneously. The central ideas are to employ a flexible set of nonlinear prediction functions and to prevent overfitting by penalization, in a way that limits the computational complexity. In this approach, which is known as *kernel ridge regression*, the set of predictors is mapped into a high-dimensional (or even infinite-dimensional) space of nonlinear functions of the predictors. A forecast equation is esti-

mated in this high-dimensional space, using a penalty (or shrinkage, or ridge) term to avoid overfitting. In this manner, kernel ridge regression does not suffer from the curse of dimensionality, which plagues alternative nonparametric approaches to allow for flexible types of nonlinearity (Pagan and Ullah, 1999). Computational tractability is achieved by choosing the kernel in a convenient way, so that calculations in the high-dimensional space actually are prevented. This approach avoids computational difficulties also encountered in standard linear ridge regression when the number of predictor variables is large relative to the number of time series observations. Taking all these elements together, kernel ridge regression provides an attractive framework for estimating nonlinear predictive relations in a data-rich environment.

The kernel methodology has been developed in the machine learning community, an area which often involves large data sets. The terminology originates from operator theory, as computations are performed in terms of the kernel of a positive integral operator, see Vapnik (1995). We use the term *kernel* in this sense, as it is the established term for this method in machine learning. This meaning should not be confused with other uses of the word, such as in kernel smoothing methods for local regression.

A typical application of kernel methods is classification, for example, in optical recognition of pixel-by-pixel scans of handwritten characters. Schölkopf et al. (1998) document outstanding performance of kernel methods for this classification task. Kernel ridge regression has been found to work well also in many other applications. Time-series applications are scarce and seem to be limited to deterministic (that is, non-stochastic) time series (Müller et al., 1997). Kernel ridge regression has, to our knowledge, not yet been applied in the context of macroeconomic or financial time-series forecasting.

In this paper, we provide two methodological contributions to kernel ridge regression. First, we extend the approach to enable the use of models that include lags of the dependent variable or other individual variables as predictors, as is typically desired in such applications. Second, we derive an efficient cross-validation procedure for selecting the tuning parameters involved in kernel ridge regression, in particular, the shrinkage parameter that determines the strength of the penalization factor.

We provide simulation evidence, demonstrating that kernel ridge regression delivers more accurate forecasts than conventional methods based on principal components in the presence of many predictors that are related nonlinearly with the target variable. These conventional methods include extensions of principal component regression to accommodate nonlinearity as put forward by Bai and Ng (2008). The potential practical usefulness of kernel methods is confirmed in an empirical application to forecasting four key measures of U.S. macroeconomic activity over the period 1970-2009: Industrial Production,

Personal Income, Manufacturing & Trade Sales, and Employment. We find that, when traditional methods perform poorly, kernel ridge regression yields substantial improvements. This result holds for the Production and Income series. When traditional forecasts are of good quality, as is the case for the Sales and Employment series, kernel-based forecasts remain competitive. We also find that kernel ridge regression is much less affected by the 2008-9 financial and economic crisis than traditional methods.

The remainder of this paper is organized as follows. Section 2 describes the kernel methodology. The Monte Carlo simulation is presented in Section 3, and Section 4 discusses the empirical application. Conclusions are provided in Section 5. Details of the technical results are collected in an Appendix.

2 Methodology

The technique of kernel ridge regression (KRR) is based on ordinary least squares (OLS) regression and ridge regression. Therefore, we begin this section with a brief review of these methods, highlighting their drawbacks in dealing with nonlinearity and high-dimensionality. Next, we show how kernel ridge regression overcomes these drawbacks by means of the so-called *kernel trick*. We extend the KRR methodology to allow for "preferred" predictors, to enable its use in time-series contexts. We also present the properties of some kernel functions that are popular because of their computational efficiency. As will become clear below, kernel ridge regression involves tuning parameters. In Section 2.4 we propose an efficient cross-validation procedure for selecting values for these parameters.

2.1 Preliminaries

Consider the following general setup for forecasting. At the end of period T we wish to forecast the value of a target variable y at a specific future date, denoted y_*, given an $N \times 1$ vector of predictors x_*. Historical observations for $t = 1, \ldots, T$ are available for all variables, collected in the $T \times 1$ vector y and the $T \times N$ matrix X. If we assume a linear prediction function $\hat{y}_* = x_*'\hat{\beta}$, we may obtain $\hat{\beta}$ by minimizing the OLS criterion $||y - X\beta||^2$, where $||\cdot||$ denotes the L_2 norm. Provided that X has rank N, the solution is $\hat{\beta} = (X'X)^{-1} X'y$, which leads to the forecast $\hat{y}_* = x_*' (X'X)^{-1} X'y$.

The OLS procedure presupposes that $N \leq T$, and in practice, $N \ll T$ is required to prevent overfitting problems. That is, if N is not small compared to T, we may obtain a good in-sample fit (indeed, if $N = T$, the in-sample fit will be perfect), but the out-of-sample prediction $\hat{y}_*$ is generally found to be of poor quality. A possible solution to this problem is shrinkage estimation or ridge regression, which

aims to balance the goodness-of-fit and the magnitude of the coefficient vector β. The ridge criterion is given by $||y - X\beta||^2 + \lambda ||\beta||^2$, where the penalty parameter $\lambda > 0$ is to be specified by the user. As every element of the parameter vector β is equally penalized, the predictors in X should be scaled appropriately. In our applications, we studentize each column of X over the estimation sample, so that each predictor has zero mean and unit variance. The solution $\hat{\beta}$ that minimizes the ridge criterion is most easily found by defining the $(T+N) \times 1$ vector $u = \left(y', 0'_{N\times 1}\right)'$ and the $(T+N) \times N$ matrix $V = \left(X', \sqrt{\lambda}\, I_{N\times N}\right)'$, where $I_{N\times N}$ denotes the N-dimensional identity matrix. We may then write $||y - X\beta||^2 + \lambda ||\beta||^2 = ||u - V\beta||^2$. Minimizing this criterion by OLS yields $\hat{\beta} = (V'V)^{-1} V'u$, or, in terms of the original variables, $\hat{\beta} = (X'X + \lambda I)^{-1} X'y$ (we omit the subscript $N \times N$ from I for notational convenience). The resulting forecast $\hat{y}_* = x'_* (X'X + \lambda I)^{-1} X'y$ can be computed also if the number of predictors N is larger than the number of observations T. Nevertheless, if N becomes very large, the calculation of the ridge forecast may present computational difficulties, as it involves inverting the $N \times N$ matrix $X'X + \lambda I$. In practice, this hampers the use of ridge regression when $N \gg T$.

2.2 Kernel ridge regression and the kernel trick

Kernel ridge regression extends the general setup considered above to allow for nonlinear prediction functions $\hat{y}_* = f(x_*)$. At the same time, it provides a way to avoid the computational complications involved in producing the ridge forecast when the number of predictors becomes very large. As will become clear below, this is particularly relevant in the context of nonlinear forecasting. From now on, let $\varphi : \mathbb{R}^N \to \mathbb{R}^M$ be a (possibly nonlinear) mapping of the N observed predictor variables x resulting in M transformed predictor variables $z = \varphi(x)$. We assume that the prediction function is linear in z, say $\hat{y}_* = z'_* \hat{\gamma}$, where $z_* = \varphi(x_*)$. Collecting the transformed predictor variables in the $T \times M$ matrix Z with rows $z'_t = \varphi(x_t)'$, we may apply ridge regression to obtain $\hat{\gamma} = (Z'Z + \lambda I)^{-1} Z'y$, and hence,

$$\hat{y}_* = z'_* \left(Z'Z + \lambda I\right)^{-1} Z'y. \tag{1}$$

In macroeconomic and financial applications we often work with high-dimensional data, sometimes with the number of observed predictors N exceeding the number of time series observations T. Moreover, to allow for flexible forms of nonlinearity in the forecast equation, we need $M \gg N$. For example, if we approximate the unknown forecast function f by a dth order Taylor expansion, the mapping φ effectively transforms the $N \times 1$ vector x into the $M \times 1$ vector z containing powers and cross-products

of its elements, with M proportional to N^d. Thus, M may become extremely large for realistic values of N and d. As the matrix $Z'Z$ has dimensions $M \times M$, this can cause computational difficulties.

An efficient method to solve this curse of dimensionality problem is provided by the so-called kernel trick. Essentially this method is based on the idea that if the number of regressors M is much larger than the number of observations T, working with T-dimensional instead of M-dimensional objects can lead to notable computational savings. To appreciate the dimension reductions involved, we consider the macroeconomic application that will be discussed in Section 4. In this application, we estimate models with $N = 132$ predictors on an estimation sample containing $T = 120$ observations. One of the models includes a constant, all observed predictors, their squares, and all pairwise cross-products, leading to a total of $M = (N+1)(N+2)/2 = 8911$ transformed predictor variables. The results described in the remainder of this section allow working with a 120×120 matrix instead of the 8911×8911 matrix $Z'Z$, leading to sizeable computational savings. What is more, as we shall see in Section 2.3, the kernel trick can also be made to work in cases with $M \to \infty$, where standard ridge regression cannot be applied.

This dimension reduction can be achieved by relatively straightforward algebraic manipulations of the expression of the nonlinear ridge forecast equation $\hat{y}_* = z_*'\hat{\gamma}$. First, we rewrite the ridge regression estimator $\hat{\gamma} = (Z'Z + \lambda I)^{-1} Z'y$ as $Z'Z\hat{\gamma} + \lambda\hat{\gamma} = Z'y$, or

$$\hat{\gamma} = \frac{1}{\lambda}\left(Z'y - Z'Z\hat{\gamma}\right) = \frac{1}{\lambda}Z'\left(y - Z\hat{\gamma}\right).$$

If we pre-multiply $Z'Z\hat{\gamma} + \lambda\hat{\gamma} = Z'y$ by the matrix Z, this gives $ZZ'Z\hat{\gamma} + \lambda Z\hat{\gamma} = ZZ'y$, or

$$Z\hat{\gamma} = \left(ZZ' + \lambda I\right)^{-1} ZZ'y.$$

Combining these two results, we find

$$\begin{aligned}\hat{y}_* &= z_*'\hat{\gamma} = \frac{1}{\lambda}z_*'Z'\left(y - Z\hat{\gamma}\right) = \frac{1}{\lambda}z_*'Z'\left(y - \left(ZZ' + \lambda I\right)^{-1} ZZ'y\right) \\ &= \frac{1}{\lambda}z_*'Z'\left(ZZ' + \lambda I\right)^{-1}\left(ZZ' + \lambda I - ZZ'\right)y = z_*'Z'\left(ZZ' + \lambda I\right)^{-1} y.\end{aligned}$$

If we define the $T \times T$ matrix $K = ZZ'$ and the $T \times 1$ vector $k_* = Zz_*$, this result can be written as

$$\hat{y}_* = k_*'\left(K + \lambda I\right)^{-1} y. \tag{2}$$

The forecast $\hat{y}_*$ in (2) is identical to the one in (1). The advantage of using (2) is that the inverse matrix in this equation has dimensions $T \times T$, so that the $M \times M$-dimensional computations in (1) are prevented.

To achieve computational savings over the straightforward application of ridge regression, it is crucial that K and k_* can be computed in a relatively simple way. The (s,t)-th element of $K = ZZ'$ equals $z_s'z_t = \varphi(x_s)'\varphi(x_t)$, and similarly, the t-th element of k_* equals $\varphi(x_t)'\varphi(x_*)$. This implies that the computational efficiency increases greatly if we choose a mapping φ for which the inner product $\kappa(a,b) = \varphi(a)'\varphi(b)$ can be computed quickly, that is, without computing $\varphi(a)$ and $\varphi(b)$ explicitly. In this context, κ is called the *kernel function* and K is the *kernel matrix*. This procedure for implicitly finding the optimal parameter vector $\hat{\gamma}$ in the "high" dimension M while working exclusively in the "low" dimension T is known as the *kernel trick* and is due to Boser et al. (1992).

As the above discussion shows, KRR is no different from ordinary ridge regression on transformations of the regressors, except for an algebraic trick to improve computational efficiency. The key to a successful application of this kernel trick is choosing a mapping φ that leads to an easy-to-compute kernel function κ, while, obviously, at the same time φ should be chosen such that the corresponding prediction function $\varphi(x_*)'\gamma$ provides a good approximation to the true but unknown nonlinear prediction function $f(x_*)$. Various such mappings are known, and a recent overview is given in Smola and Schölkopf (2004). The next section presents the most commonly used instances of these mappings.

In a time series context, we often prefer to include specific predictors in the forecast equation separately from the nonlinear mapping φ. In macroeconomic applications, these "preferred" predictors may include lags of the dependent variable to account for serial correlation. In financial applications such as predicting stock returns, these predictors may include valuation ratios such as the dividend yield or interest rate related variables; see Ludvigson and Ng (2007), Çakmaklı and van Dijk (2010), for example. In such cases the generalized forecast equation takes the form $\hat{y}_* = w_*'\hat{\beta} + z_*'\hat{\gamma}$, where the $P \times 1$ vector w_* contains the variables to be treated linearly. As the number of these additional predictors is limited and the effect of these predictors is of particular interest, we do not penalize the parameters β and restrict the ridge penalization to γ. We show in Appendix A.1 that the derivations that lead to (2) can be extended to include such linear unpenalized terms, resulting in the "extended" KRR forecast equation

$$\hat{y}_* = \begin{pmatrix} k_* \\ w_* \end{pmatrix}' \begin{pmatrix} K + \lambda I & W \\ W' & 0 \end{pmatrix}^{-1} \begin{pmatrix} y \\ 0 \end{pmatrix}, \tag{3}$$

where the $T \times P$ matrix W contains the historical observations on the variables to be treated linearly. This is the forecast equation that will be used in the empirical application in Section 4.

2.3 Some common kernel functions

A first and obvious example is the identity mapping $\varphi(a) = a$, for which $\kappa(a,b) = a'b$. With this choice of κ, the kernel forecast $\hat{y}_* = k_*'(K+\lambda I)^{-1} y = x_*'X'(XX'+\lambda I)^{-1} y$ equals the linear ridge forecast $\hat{y}_* = x_*'(X'X+\lambda I)^{-1} X'y$, as can be seen by taking $Z = X$ and $z_* = x_*$ in the derivations leading to (2).

Next we consider a mapping such that $\varphi(a)$ contains a constant term, all variables $a_1, a_2, \ldots, a_N$, and all their squares and cross products. Some experimentation reveals that $\kappa(a,b)$ takes a particularly simple form if we multiply some elements of $\varphi(a)$ by the constant $\sqrt{2}$. That is, if we choose the mapping

$$\varphi(a) = \left(1, \sqrt{2}a_1, \sqrt{2}a_2, \ldots, \sqrt{2}a_N, a_1^2, a_2^2, \ldots, a_N^2, \sqrt{2}a_1a_2, \sqrt{2}a_1a_3, \ldots, \sqrt{2}a_{N-1}a_N\right)',$$

the corresponding kernel function is

$$\begin{aligned}
\kappa(a,b) &= \varphi(a)'\varphi(b) \\
&= 1 + 2(a_1b_1 + a_2b_2 + \ldots + a_Nb_N) + a_1^2b_1^2 + a_2^2b_2^2 + \ldots + a_N^2b_N^2 \\
&\quad + 2(a_1a_2b_1b_2 + a_1a_3b_1b_3 + \ldots + a_{N-1}a_Nb_{N-1}b_N) \\
&= 1 + 2(a_1b_1 + a_2b_2 + \ldots + a_Nb_N) + (a_1b_1 + a_2b_2 + \ldots + a_Nb_N)^2 \\
&= 1 + 2a'b + (a'b)^2 \;=\; (1+a'b)^2
\end{aligned}$$

With this specification of the kernel function, the computation of each of the $T(T+1)/2$ distinct elements of the kernel matrix K requires $2(N+1)$ additions and multiplications. In the absence of the indicated scaling, the vector of constant, first-order, and second-order terms contains $M = (N+1)(N+2)/2$ elements. The computation of each element of the kernel matrix would then require $2M = (N+1)(N+2)$ additions and multiplications. Thus, the proposed scaling reduces the amount of computations by a factor of $(N+2)/2$.

As noted by Poggio (1975), this result can be generalized to the kernel function

$$\kappa(a,b) = (1+a'b)^d \qquad \text{for any integer } d \geq 1, \tag{4}$$

corresponding to a mapping for which $\varphi(a)$ consists of all polynomials in the elements of a of degree at most d. Observe that this class of so-called polynomial kernel functions encompasses not only the quadratic mapping, for $d = 2$, but also the identity mapping (and hence, standard linear ridge regression), for $d = 1$.

Because smart choices of φ enable us to avoid M-dimensional computations, the kernel methodology even allows letting $M \to \infty$. A common way to do this, dating back to Broomhead and Lowe (1988), is by using the Gaussian kernel function

$$\kappa(a, b) = \exp\left(-\frac{1}{2}||a - b||^2\right). \tag{5}$$

We show in Appendix A.2 that the corresponding mapping $\varphi(a)$ contains as elements, for all degrees $d_1, d_2, \ldots, d_N \geq 0$, the "dampened" polynomials

$$e^{-a'a/2} \prod_{n=1}^{N} \frac{a_n^{d_n}}{\sqrt{d_n!}}.$$

In this paper, we consider the polynomial kernels (4) of degrees $d = 1$ and 2, as well as the Gaussian kernel (5). To control for the relative importance of the terms in $\varphi(x)$, we replace each observation x by $(1/\sigma)x$ before computing κ, for some positive scaling factor σ. Such scaling affects the weight placed on different polynomial degrees, as it amounts to dividing linear terms in $\varphi(x)$ by σ, second-order terms by σ^2, and so forth. Although we are performing linear regression on $\varphi(x)$, such scaling is not without effect, as its regression coefficients in the forecast equation $\hat{y}_* = w_*'\hat{\beta} + \varphi(x_*)'\hat{\gamma}$ are all penalized equally by the ridge term in the criterion function $||y - W\beta - Z\gamma||^2 + \lambda||\gamma||^2$.

2.4 Selection of tuning parameters

The implementation of kernel ridge regression involves two tuning parameters, namely, the shrinkage parameter λ and the scaling parameter σ. Additionally, our empirical application in Section 4 involves the selection of lag lengths, which can also be seen as tuning parameters from a model selection perspective. This section addresses the question of how to select the values for these tuning parameters.

We determine the values of the tuning parameters by means of leave-one-out cross-validation, as this is a natural criterion for the purpose of out-of-sample forecasting. For given values of the tuning parameters, we estimate the model on the sample of size $T - 1$ that remains when the observation for period t is removed. We then use this model to "forecast" the value of y_t that was left out. This is repeated T times, leaving out each observation for $t = 1, 2, \ldots, T$ once. Performing this cross-validation exercise for each of the candidate values of the tuning parameters, we select those values that lead to the smallest mean squared prediction error (MSPE) over these T forecasts. These values are then used to estimate the model on the full sample $t = 1, 2, \ldots, T$, from which we produce out-of-sample forecasts.

In the form stated above, this cross-validation procedure is computationally very expensive, as it requires estimating the model on T different samples for each possible setting of the tuning parameters. Cawley and Talbot (2008) propose a method that yields all leave-one-out prediction errors as a by-product of estimating (2) only once, that is, on the full sample. We derive a similar result, extended to allow for the additional linear terms in (3), in Appendix A.3.

In the simulation study and in the empirical application below, we use this method to select both lag lengths, the ridge parameter λ, and the scaling parameter σ from a grid. For the lag lengths, we employ the grids specified by Stock and Watson (2002). For the KRR parameters, we construct five-point grids based on an estimate of the signal-to-noise ratio (for λ) and a smoothness assumption (for σ). A detailed motivation and description of these grids is given by Exterkate (2012). We will use a rolling window of fixed length for estimation, and we reselect the values of the tuning parameters for each window.

As a technical note on cross-validation, serial correlation in time-series data leads to dependence between the observations in the estimation sample and the observation that was left out. This dependence implies that the standard leave-one-out cross-validation procedure may not be fully adequate; see Racine (2000) for an extensive discussion and a modification to overcome these problems. Although the method outlined in Appendix A.3 can easily be adapted to this modified form of cross-validation, the resulting implementation is computationally quite intensive. We find that the results from using this modified procedure are not appreciably different from those obtained with simple leave-one-out cross-validation (details of these results are available upon request). Therefore, we will only report the results that are obtained using the latter method.

3 Monte Carlo simulation

To evaluate the potential of kernel ridge regression in a data-rich environment (that is, when many predictor variables are present), we assess its forecasting performance for a set of static factor models through a Monte Carlo study. We consider a setting with two latent factors f_{1t} and f_{2t}, which are taken to be uncorrelated standard normal variables. As predictor variables, $N = 100$ noisy linear combinations of these factors are generated by $x_{it} = \theta_{i1} f_{1t} + \theta_{i2} f_{2t} + \eta_{it}$, where the factor loadings θ_{ij}, $j = 1, 2$, are drawn from the standard normal distribution. The noise η_{it} is also normal with mean zero, while its variance is selected to control the fraction of the variance of each x_i variable explained by the factors, denoted by

R_x^2. We consider two cases with R_x^2 equal to 0.4 or 0.8, which we label as "weak" and "strong" factor structure, respectively. The target variable y is constructed according to three different DGPs:

$$\text{Linear:} \qquad y_t = f_{1t} + f_{2t} + \varepsilon_t \tag{6}$$

$$\text{Squared:} \qquad y_t = f_{1t} + f_{2t} + 2\left(f_{1t}^2 + f_{2t}^2\right) + \varepsilon_t \tag{7}$$

$$\text{Cross-product:} \quad y_t = f_{1t} + f_{2t} + 4 f_{1t} f_{2t} + \varepsilon_t \tag{8}$$

Here ε_t is normal with mean zero and a variance selected to control R_y^2, the fraction of the variance of y_t that is explained by the factors. For R_y^2 we also consider the values 0.4 and 0.8, which we refer to as "weak" and "strong" predictive structure, respectively.

In each Monte Carlo replication, we generate time series of x_i, $i = 1, \ldots, N$, and y, each consisting of $T + 1$ observations. The first T observations are used for estimation, and a forecast for y_{T+1} is made based on x_{T+1}. All variables are standardized to have mean zero and variance one in the estimation sample. We set $T = 120$, which corresponds to the length of each estimation window (ten years of monthly observations) in the empirical application in Section 4. We present results based on 5000 replications.

In principle, OLS regression using the individual predictors can be applied in this setting, as the number of regressors is smaller than the number of observations in the estimation sample. It should not come as a surprise, though, that this procedure leads to a very poor forecasting performance, given the large amount of parameter estimation uncertainty when $N = 100$ and $T = 120$. We therefore do not report OLS results. Instead, we consider four alternative prediction methods for comparison with KRR:

(i) the "mean" forecast, with $\hat{y}_{121} = (1/120) \sum_{t=1}^{120} y_t$;

(ii) principal component regression (PC), which amounts to OLS but with regressors $\hat{f}_t$ being the first k principal components of the predictor variables x;

(iii) "PC-squared" (PC^2), as suggested by Bai and Ng (2008), which corresponds to principal component regression with the squares of $\hat{f}_t$ as additional regressors; and

(iv) "Squared PC" (SPC), also proposed by Bai and Ng (2008), which is principal component regression but using the principal components of the original predictor variables x and their squares.

Bai and Ng (2008) also propose a quadratic principal component (QPC) regression variant, in which principal components are taken not only of the x_{it} and their squares (as in SPC), but also including their

Table 1: Relative mean squared prediction errors for the factor models (6)-(8).

DGP	Linear				Squared				Cross-product			
	$R_y^2 = 0.4$		$R_y^2 = 0.8$		$R_y^2 = 0.4$		$R_y^2 = 0.8$		$R_y^2 = 0.4$		$R_y^2 = 0.8$	
$R_x^2 =$	0.4	0.8	0.4	0.8	0.4	0.8	0.4	0.8	0.4	0.8	0.4	0.8
Mean	1.00	1.00	1.02	1.02	1.02	1.02	1.07	1.07	1.03	1.03	1.08	1.08
PC	**0.62**	**0.61**	**0.23**	**0.20**	1.01	1.01	1.02	1.02	1.03	1.03	1.04	1.04
PC^2	0.63	0.62	0.23	0.21	**0.65**	**0.63**	**0.27**	**0.22**	0.88	0.87	0.68	0.66
SPC	0.63	0.63	0.24	0.22	0.69	0.64	0.34	0.22	0.78	0.65	0.49	0.23
Poly(1)	0.65	0.62	0.24	0.21	1.01	1.01	1.02	1.01	1.03	1.03	1.05	1.04
Poly(2)	0.65	0.63	0.25	0.21	0.71	0.64	0.33	0.23	**0.71**	**0.65**	**0.34**	**0.23**
Gauss	0.65	0.63	0.25	0.22	0.80	0.71	0.51	0.34	0.87	0.78	0.65	0.44

Notes: This table reports mean squared prediction errors (MSPEs) for models (6)-(8), averaged over 5000 forecasts, and relative to the variance of the series being predicted. The smallest relative MSPE for each DGP (column) is printed in boldface.

cross-products. They report high computational costs and poor forecasting performance for QPC, and our preliminary analysis confirms these results. For this reason, QPC is not considered in our study.

For KRR, the tuning parameters λ and σ are selected from the grids defined in Section 2.4 using leave-one-out cross-validation. For each of the principal-components-based methods, we select the number of components k by minimizing the Bayesian Information Criterion (BIC), where $1 \leq k \leq 10$. Our reason for minimizing BIC instead of performing cross-validation for these methods is twofold. First, using BIC in principal components forecasting settings is common in the literature; see, for example, Stock and Watson (2002) and Bai and Ng (2008). Second, preliminary simulation evidence shows that using BIC leads to superior results compared to using cross-validation.

Table 1 shows mean squared prediction errors (MSPEs) relative to the variance of the series being predicted. Note that if the factor values $f_{1,T+1}$ and $f_{2,T+1}$ were known, these relative MSPEs would be close to $1 - R_y^2$, or 0.6 and 0.2 in the two scenarios of "weak" ($R_y^2 = 0.4$) and "strong" ($R_y^2 = 0.8$) predictive structure considered here. Standard PC shows good performance for the linear DGP, while PC^2 performs well for the squared DGP. Such results were to be expected, because the forecast equation in these methods corresponds exactly with these DGPs. Interestingly, the kernel methods are not much less accurate than these "optimal" methods, with the obvious exception of the Poly(1) (that is, linear) kernel in the squared DGP (for which standard PC also fares badly). This finding holds regardless of whether R_x^2 and R_y^2 are high or low, although the difference between PC or PC^2 and the best performing kernel method is smaller when the factor structure in the predictor variables is stronger (compare $R_x^2 = 0.4$ with $R_x^2 = 0.8$). Thus, we find that kernel methods can work well in standard factor model settings, even though these methods are not based on any factor model assumptions.

For the cross-product DGP, the SPC method from Bai and Ng (2008) and the Poly(2) kernel can both be expected to perform well. We observe that KRR provides the most accurate forecasts here, and that the gains are larger for lower R_x^2. Thus KRR performs well in this case, especially when the factor structure of the predictors is not very strong, as is often the case for empirical macroeconomic and financial data. The Gaussian kernel also performs reasonably well.

We conclude that kernel methods work quite well in a factor context, especially for nonlinear relations and in situations where the observed predictors give relatively little information on the factors.

4 Macroeconomic forecasting

4.1 Data and forecast models

We evaluate the forecast performance of kernel ridge regression in an empirical application, involving a large panel of U.S. macroeconomic and financial variables. The data set consists of monthly observations on 132 variables, including various measures of production, consumption, income, sales, employment, monetary aggregates, prices, interest rates, and exchange rates. All series are transformed to stationarity by taking logarithms and/or differences, as described in Stock and Watson (2005). We have updated their data set, which starts in January 1959 and ends in December 2003, to cover the period up to (and including) January 2010. The cross-sectional dimension varies somewhat over time because of data availability: some time series start later than January 1959, while a few other variables have been discontinued before the end of our sample period. For each month under consideration, observations on at most five variables are missing.

We focus on forecasting four key measures of real economic activity: Industrial Production, Personal Income less Transfer Payments (referred to as Personal Income in the following), Manufacturing & Trade Sales, and Employment on Non-Agricultural Payrolls (referred to as Employment in the remainder of this section), as in Stock and Watson (2002), among others. For each of these variables, we produce out-of-sample forecasts for the annualized h-month percentage growth rate, computed as

$$y_{t+h}^h = \frac{1200}{h} \ln\left(\frac{v_{t+h}}{v_t}\right),$$

where v_t is the untransformed observation on the level of each variable in month t. We will denote y_{t+1}^1 as y_{t+1} to simplify the notation. We consider growth rate forecasts for $h = 1, 3, 6$ and 12 months, and

we follow Stock and Watson (2002) in modelling the h-month growth rate directly, rather than making iterated one-month-ahead forecasts.

Kernel ridge regression is compared against several alternative forecasting approaches that are popular in current macroeconomic practice. As benchmarks we include the "mean" forecast (that is, the average growth over the estimation window), the "no-change" or random-walk (RW) forecast, and an autoregressive (AR) forecast (using lagged values of the one-month growth rates as predictors). In addition, as the primary competitor for kernel methods we consider the diffusion index (DI) approach of Stock and Watson (2002), who document its good performance for forecasting the same four macroeconomic variables as considered here. The DI methodology extends the standard principal component regression to a dynamic setting by including autoregressive lags as well as lags of the principal components in the forecast equation. Specifically, using p autoregressive lags and q lags of k factors, at time t, this "extended" principal-components method produces the forecast

$$\hat{y}^h_{t+h|t} = w_t'\hat{\beta} + \hat{f}_t'\hat{\gamma},$$

where $w_t = \left(1, y_t, y_{t-1} \ldots, y_{t-(p-1)}\right)'$ and $\hat{f}_t = \left(\hat{f}_{1,t}, \hat{f}_{2,t}, \ldots, \hat{f}_{k,t}, \hat{f}_{1,t-1}, \ldots, \hat{f}_{k,t-(q-1)}\right)'$. The lags of the dependent variable in w_t are one-month growth rates, irrespective of the forecast horizon h, because using h-month growth rates for $h > 1$ would lead to highly correlated regressors. The factors $\hat{f}$ are principal components extracted from all 132 predictor variables, and $\hat{\beta}$ and $\hat{\gamma}$ are OLS estimates. Aside from standard principal components (PC), we also consider its extensions PC^2 and SPC, discussed in Section 3. In each case, the lag lengths p and q and the number of factors k are selected by minimizing the Bayesian Information Criterion (BIC). This criterion is used instead of cross-validation for two reasons. We want our results to be comparable to those in Stock and Watson (2002) and Bai and Ng (2008), and preliminary experimentation with the PC methods has revealed that using the BIC leads to superior results. Following Stock and Watson (2002), we allow $0 \leq p \leq 6$ (where $p = 0$ means that $w_t = 1$), $1 \leq q \leq 3$, and $1 \leq k \leq 4$. Thus, the simplest model that can be selected uses no information on current or lagged values of the dependent variable, and information from the other predictors in the current month only, summarized by a single factor. Also in line with Stock and Watson (2002), we do not perform an exhaustive search across all possible combinations of the first four principal components and lag structures. Instead, we assume that factors are included sequentially in order of importance, while the number of lags is assumed to be the same for all included factors.

For KRR, the corresponding forecast equation is

$$\hat{y}^h_{t+h|t} = w_t'\hat{\beta} + \varphi\left(\left(x_t', x_{t-1}', \ldots, x_{t-(q-1)}'\right)'\right)'\hat{\gamma},$$

in the notation of Section 2.2, where w_t is as defined above and x_t contains all 132 predictors at time t. The parameter vectors $\hat{\beta}$ and $\hat{\gamma}$ are obtained by KRR, resulting in the forecast equation (3). In particular, note that β (which contains the constant term and the autoregressive coefficients) is not subject to a ridge penalty, in order to avoid biased estimation of this short vector of relatively important parameters. The lag lengths p and q, as well as the KRR parameters λ and σ, are selected by leave-one-out cross-validation.

All models are estimated on rolling windows with a fixed length of 120 months, such that the first forecast is produced for the growth rate during the first h months of 1970. For each window, the tuning parameter values are re-selected and the regression coefficients are re-estimated. That is, all of the tuning parameters $(p, q, k, \lambda, \sigma)$ may differ over time and across target variables, horizons, and methods.

4.2 Results

Table 2 shows the MSPEs for the period 1970-2010 for the three benchmark methods, three PC-based methods, and three kernel methods. Several conclusions can be drawn from these results. First, we observe that KRR provides more accurate forecasts than any of the three benchmarks (mean, random walk, and autoregression) for all target variables and all forecast horizons, with larger gains for longer horizons. This holds irrespective of which kernel function is used. In many cases the improvements in predictive accuracy are substantial, even compared to the AR forecast, which seems the best of the three benchmarks. For example, for 12-month growth rate forecasts, kernel ridge regression based on the Gaussian kernel achieves reductions in MSPE of 30-40% for all four variables (relative to AR).

Second, if we compare the forecasts based on KRR and the linear PC-based approach, we find somewhat mixed results, but generally the kernel methods perform better. Kernel ridge forecasts are superior for Industrial Production and Personal Income for three of the four horizons considered. The improvements in relative MSPE range from 0.03 for Personal Income at the 6-month horizon to 0.11, also for Personal Income at the shortest horizon of one month. For Manufacturing & Trade Sales, kernels perform slightly worse than linear principal components, but the difference is small especially at the longer horizons. Finally, for Employment, the PC-based forecasts are more accurate than kernel-based forecasts by about 20%.

Table 2: Relative mean squared prediction errors for the macroeconomic series.

Forecast	Industrial Production				Personal Income			
method	$h=1$	$h=3$	$h=6$	$h=12$	$h=1$	$h=3$	$h=6$	$h=12$
Mean	1.02	1.05	1.07	1.08	1.02	1.06	1.10	1.17
RW	1.27	1.08	1.34	1.64	1.60	1.36	1.14	1.35
AR	0.93	0.89	1.02	1.02	1.17	1.05	1.10	1.15
PC	0.81	0.71	0.77	**0.63**	1.04	**0.79**	0.90	0.90
PC^2	0.94	0.85	1.20	1.07	1.09	0.92	1.03	1.15
SPC	0.88	0.98	1.35	0.99	1.07	1.04	1.05	1.50
Poly(1)	0.79	0.73	0.75	0.68	0.98	0.88	0.89	0.91
Poly(2)	0.79	0.72	0.80	0.68	0.97	0.85	0.93	0.96
Gauss	**0.76**	**0.66**	**0.73**	0.66	**0.93**	0.83	**0.87**	**0.85**

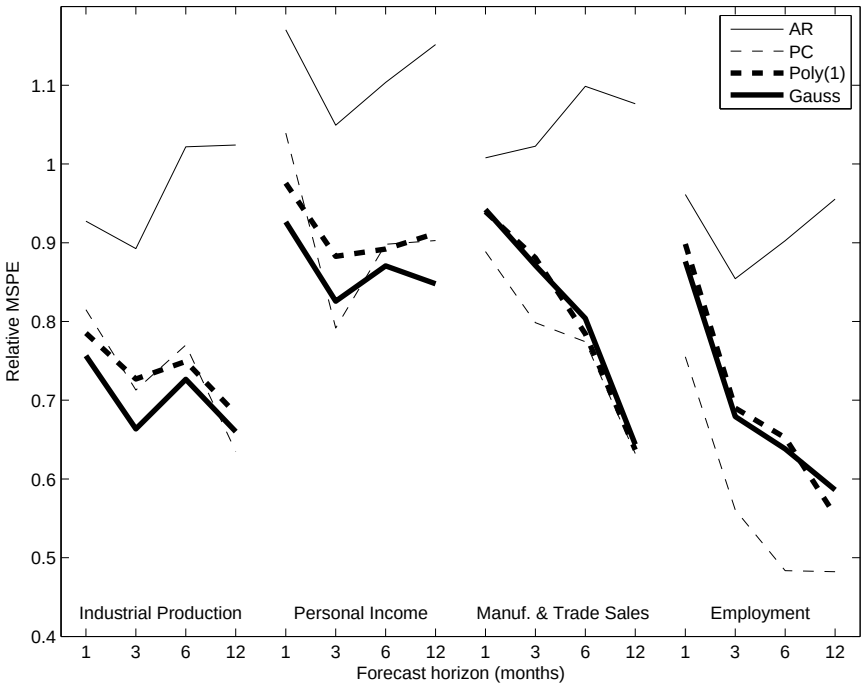
Forecast	Manufacturing & Trade Sales				Employment			
method	$h=1$	$h=3$	$h=6$	$h=12$	$h=1$	$h=3$	$h=6$	$h=12$
Mean	1.01	1.03	1.05	1.08	0.98	0.96	0.97	0.97
RW	2.17	1.49	1.45	1.53	1.68	0.95	1.00	1.20
AR	1.01	1.02	1.10	1.08	0.96	0.85	0.90	0.96
PC	**0.89**	**0.80**	**0.77**	**0.63**	**0.76**	**0.56**	**0.48**	**0.48**
PC^2	0.94	0.97	1.13	1.06	0.76	0.61	0.69	0.60
SPC	0.99	1.18	1.59	1.02	0.81	0.81	0.90	0.72
Poly(1)	0.94	0.88	0.78	0.64	0.90	0.69	0.65	0.55
Poly(2)	0.96	0.88	0.81	0.67	0.95	0.70	0.69	0.64
Gauss	0.94	0.87	0.80	0.64	0.88	0.68	0.64	0.59

Notes: This table reports mean squared prediction errors (MSPEs) for four macroeconomic series, over the period 1970-2010, relative to the variance of the series being predicted. The smallest relative MSPE for each series (column) is printed in boldface.

Third, the KRR approach convincingly outperforms the PC^2 and SPC variants of the principal component regression framework. In fact, also the linear PC specification renders substantially more accurate forecasts than these two extensions in all cases. Apparently, the PC^2 and SPC methods cannot successfully cope with the possibly nonlinear relations between the target variables and the predictors in this application. (Bai and Ng (2008) report somewhat better forecast performance if SPC is applied to a selected subset of the predictors, rather than to the full predictor set. Also with this modification, SPC has difficulties outperforming simpler linear methods in their application.)

Fourth, among the kernel-based methods, the Gaussian kernel generally performs best, achieving lower MSPEs than either polynomial kernel in all but a few cases. Although Poly(1) — that is, linear ridge regression — performs better than the Gaussian kernel in a few cases, the latter kernel method shows satisfactory results in all situations. Furthermore, all MSPE / variance ratios in Table 2 are below one for all kernels.

Figure 1: Relative mean squared prediction errors for four macroeconomic series, for selected methods.



A subset of the results in Table 2 is reproduced graphically in Figure 1. This graph allows us to interpret the mixed results in the comparison of kernel-based and linear PC-based forecasts as follows. KRR (especially using the Gaussian kernel) shows roughly the same good performance for all four series. However, the quality of PC forecasts varies substantially among the series and is exceptionally high for the Employment series. Recall that in the Monte Carlo experiment in Section 3, we find the analogous result that kernel-based methods yield better relative performance, compared to PC-based methods, if the factor structure is relatively weak. That is, our results suggest that kernel ridge regression performs better than principal component regression unless the latter performs very well.

Following Stock and Watson (2002), we provide a further evaluation of our results by using the forecast combining regression

$$y_{t+h}^h = \alpha\, \hat{y}_{t+h|t}^h + (1-\alpha)\, \hat{y}_{t+h|t}^{h,\,\mathrm{AR}} + u_{t+h}^h, \tag{9}$$

where y_{t+h}^h is the realized growth rate over the h-month period ending in month $t+h$, $\hat{y}_{t+h|t}^h$ is a candidate forecast from either the PC-based methods or from kernel ridge regression made at time t, and $\hat{y}_{t+h|t}^{h,\,\mathrm{AR}}$ is the benchmark autoregressive forecast. Estimates of α are shown in Table 3, with heteroscedasticity and autocorrelation consistent (HAC) standard errors in parentheses. The null hypothesis that the AR forecast receives unit weight ($\alpha = 0$) is strongly rejected in all cases, which means that PC-based

Table 3: Estimated coefficients $\hat{\alpha}$ from the forecast combining regression (9).

Forecast	Industrial Production				Personal Income			
method	$h=1$	$h=3$	$h=6$	$h=12$	$h=1$	$h=3$	$h=6$	$h=12$
PC	0.83^{*} (0.15)	0.80^{*} (0.14)	$0.72^{*\dagger}$ (0.13)	0.79^{*} (0.11)	0.97^{*} (0.26)	0.87^{*} (0.12)	$0.70^{*\dagger}$ (0.10)	$0.70^{*\dagger}$ (0.09)
PC^2	$0.48^{*\dagger}$ (0.15)	$0.55^{*\dagger}$ (0.11)	$0.42^{*\dagger}$ (0.12)	$0.48^{*\dagger}$ (0.13)	0.71^{*} (0.18)	$0.66^{*\dagger}$ (0.12)	$0.54^{*\dagger}$ (0.07)	$0.50^{*\dagger}$ (0.10)
SPC	$0.57^{*\dagger}$ (0.08)	$0.43^{*\dagger}$ (0.11)	$0.37^{*\dagger}$ (0.12)	$0.51^{*\dagger}$ (0.08)	0.75^{*} (0.21)	$0.51^{*\dagger}$ (0.09)	$0.52^{*\dagger}$ (0.10)	$0.39^{*\dagger}$ (0.08)
Poly(1)	$0.72^{*\dagger}$ (0.09)	$0.73^{*\dagger}$ (0.13)	$0.74^{*\dagger}$ (0.12)	$0.73^{*\dagger}$ (0.11)	0.84^{*} (0.22)	$0.74^{*\dagger}$ (0.11)	$0.68^{*\dagger}$ (0.12)	$0.64^{*\dagger}$ (0.09)
Poly(2)	$0.70^{*\dagger}$ (0.09)	$0.71^{*\dagger}$ (0.12)	$0.68^{*\dagger}$ (0.11)	$0.73^{*\dagger}$ (0.11)	0.86^{*} (0.26)	$0.74^{*\dagger}$ (0.12)	$0.64^{*\dagger}$ (0.12)	$0.60^{*\dagger}$ (0.08)
Gauss	$0.79^{*\dagger}$ (0.10)	0.80^{*} (0.12)	0.77^{*} (0.12)	$0.77^{*\dagger}$ (0.11)	0.93^{*} (0.24)	0.79^{*} (0.12)	$0.69^{*\dagger}$ (0.11)	$0.69^{*\dagger}$ (0.09)
Forecast	Manufacturing & Trade Sales				Employment			
method	$h=1$	$h=3$	$h=6$	$h=12$	$h=1$	$h=3$	$h=6$	$h=12$
PC	0.83^{*} (0.12)	0.86^{*} (0.13)	0.87^{*} (0.17)	0.91^{*} (0.12)	1.02^{*} (0.09)	0.93^{*} (0.09)	0.92^{*} (0.10)	1.04^{*} (0.12)
PC^2	$0.64^{*\dagger}$ (0.08)	$0.55^{*\dagger}$ (0.11)	$0.48^{*\dagger}$ (0.19)	$0.51^{*\dagger}$ (0.15)	0.91^{*} (0.06)	$0.74^{*\dagger}$ (0.07)	$0.62^{*\dagger}$ (0.10)	0.77^{*} (0.14)
SPC	$0.53^{*\dagger}$ (0.09)	$0.39^{*\dagger}$ (0.14)	$0.29^{\dagger}$ (0.15)	$0.52^{*\dagger}$ (0.10)	$0.74^{*\dagger}$ (0.07)	$0.53^{*\dagger}$ (0.08)	$0.50^{*\dagger}$ (0.09)	$0.60^{*\dagger}$ (0.09)
Poly(1)	$0.63^{*\dagger}$ (0.10)	$0.68^{*\dagger}$ (0.13)	0.80^{*} (0.13)	0.83^{*} (0.14)	$0.61^{*\dagger}$ (0.10)	$0.72^{*\dagger}$ (0.10)	$0.75^{*\dagger}$ (0.10)	0.89^{*} (0.13)
Poly(2)	$0.57^{*\dagger}$ (0.08)	$0.69^{*\dagger}$ (0.14)	0.76^{*} (0.13)	$0.75^{*\dagger}$ (0.13)	$0.51^{*\dagger}$ (0.08)	$0.71^{*\dagger}$ (0.11)	$0.70^{*\dagger}$ (0.10)	0.76^{*} (0.13)
Gauss	$0.62^{*\dagger}$ (0.10)	$0.70^{*\dagger}$ (0.14)	0.80^{*} (0.13)	0.81^{*} (0.13)	$0.64^{*\dagger}$ (0.09)	$0.74^{*\dagger}$ (0.11)	$0.77^{*\dagger}$ (0.09)	0.85^{*} (0.13)

Notes: This table reports $\hat{\alpha}$, the weight placed on the candidate forecast in the forecast combining regression (9). HAC standard errors follow in parentheses. An asterisk ($*$) indicates rejection of the hypothesis $\alpha = 0$ and a dagger ($\dagger$) indicates rejection of $\alpha = 1$, at 5% significance.

and kernel-based forecasts have significant additional predictive ability relative to this benchmark. Actually, the null hypothesis that the candidate forecast receives unit weight ($\alpha = 1$) cannot be rejected in many cases. Note that $\alpha = 1$ in fact means that the candidate forecast encompasses the AR forecast. This hypothesis is not rejected for PC-based methods in 17 out of 48 cases, and for kernel-based methods in 14 out of 48 cases. This result confirms the conclusion drawn above that including macro factors improves forecasting performance relative to univariate benchmark models, also when allowing for nonlinear predictive relations in a flexible, nonparametric way as in kernel ridge regression.

In order to compare the performance of kernel-based and PC-based forecasts directly, we run a similar forecast combining regression

$$y_{t+h}^{h} = \alpha\, \hat{y}_{t+h|t}^{h,\,\text{KRR}} + (1-\alpha)\, \hat{y}_{t+h|t}^{h,\,\text{PC}} + u_{t+h}^{h}. \tag{10}$$

As linear PC performs better than PC^2 and SPC (see Table 2), we compare KRR to linear PC only. We report the estimates of α in Table 4. These results show that both hypotheses ($\alpha = 0$ and $\alpha = 1$) are rejected in many cases (33 out of 48), suggesting that forecasts obtained from both types of models are complementary. Apparently, each forecast method uses relevant information that the other misses.

Table 4: Estimated coefficients $\hat{\alpha}$ from the forecast combining regression (10).

Forecast	Industrial Production				Personal Income			
method	$h=1$	$h=3$	$h=6$	$h=12$	$h=1$	$h=3$	$h=6$	$h=12$
Poly(1)	$0.56^{*\dagger}$ (0.08)	$0.47^{*\dagger}$ (0.11)	$0.55^{*\dagger}$ (0.12)	$0.40^{*\dagger}$ (0.15)	0.63^{*} (0.21)	$0.25^{\dagger}$ (0.17)	$0.51^{*\dagger}$ (0.19)	$0.49^{*\dagger}$ (0.14)
Poly(2)	$0.54^{*\dagger}$ (0.07)	$0.49^{*\dagger}$ (0.11)	$0.45^{*\dagger}$ (0.11)	$0.42^{*\dagger}$ (0.12)	0.65^{*} (0.23)	$0.34^{*\dagger}$ (0.12)	$0.44^{*\dagger}$ (0.16)	$0.42^{*\dagger}$ (0.13)
Gauss	$0.63^{*\dagger}$ (0.08)	$0.60^{*\dagger}$ (0.11)	$0.61^{*\dagger}$ (0.13)	$0.44^{*\dagger}$ (0.11)	0.75^{*} (0.21)	$0.39^{*\dagger}$ (0.15)	$0.56^{*\dagger}$ (0.18)	$0.61^{*\dagger}$ (0.16)

Forecast	Manufacturing & Trade Sales				Employment			
method	$h=1$	$h=3$	$h=6$	$h=12$	$h=1$	$h=3$	$h=6$	$h=12$
Poly(1)	$0.35^{*\dagger}$ (0.11)	$0.32^{*\dagger}$ (0.10)	$0.48^{*\dagger}$ (0.16)	$0.49^{*\dagger}$ (0.20)	$0.16^{\dagger}$ (0.09)	$0.13^{\dagger}$ (0.10)	$0.04^{\dagger}$ (0.13)	$0.30^{\dagger}$ (0.17)
Poly(2)	$0.34^{*\dagger}$ (0.09)	$0.29^{*\dagger}$ (0.11)	$0.42^{*\dagger}$ (0.15)	$0.43^{*\dagger}$ (0.19)	$0.13^{\dagger}$ (0.08)	$0.09^{\dagger}$ (0.09)	-$0.04^{\dagger}$ (0.13)	$0.20^{\dagger}$ (0.15)
Gauss	$0.35^{*\dagger}$ (0.11)	$0.33^{*\dagger}$ (0.11)	$0.43^{*\dagger}$ (0.14)	$0.47^{*\dagger}$ (0.17)	$0.19^{*\dagger}$ (0.09)	$0.13^{\dagger}$ (0.10)	$0.07^{\dagger}$ (0.13)	$0.26^{\dagger}$ (0.16)

Notes: This table reports $\hat{\alpha}$, the weight placed on the kernel forecast in the forecast combining regression (10). HAC standard errors follow in parentheses. An asterisk ($*$) indicates rejection of the hypothesis $\alpha = 0$ and a dagger ($\dagger$) indicates rejection of $\alpha = 1$, at 5% significance.

Finally, we examine the stability of the performance of KRR and PC-based methods over time. For this purpose, Figure 2 shows time-series plots of rolling MSPEs for AR and for the best-performing PC and kernel methods, where the value plotted for date t is the MSPE computed over the ten-year subsample ending with the forecast for date t, that is, $\hat{y}^h_{t|t-h}$. We show only the series for $h = 12$, as the results for the other horizons are qualitatively similar. This figure confirms that, when KRR forecasts are less accurate than PC-based forecasts, this is because PC-based forecasts are very accurate, and not because KRR forecasts would be inaccurate. Another interesting feature evidenced by Figure 2 is that, although the recent crisis reduces the accuracy of all forecasts from 2008 onward, it affects the kernel-based forecasts least.

Figure 3 shows the corresponding time series of relative MSPEs for ten-year rolling windows, together with the rolling variance of the series being predicted. Together with Figure 2, these graphs lead to the following two conclusions. First, predictability improves in an absolute sense during less volatile times, in the sense that the MSPEs in Figure 2 typically decline when the rolling variance of the series being predicted in Figure 3 goes down. Second, forecasting becomes more difficult in a relative sense during less volatile periods, in the sense that the relative MSPEs seem to be inversely related to the rolling variance of the series being predicted; see Figure 3. These results corroborate the findings of Stock and Watson (2007) for U.S. inflation. Concerning the second point, it is interesting to note that the fluctuations in relative MSPE generally are more pronounced for KRR than for PC-based methods. This suggests that kernel-based forecasts are most valuable during turmoil periods.

Figure 2: Ten-year rolling-window mean squared prediction errors for four macroeconomic series, for a forecast horizon of $h = 12$ months, for AR and for the best-performing PC and kernel methods.

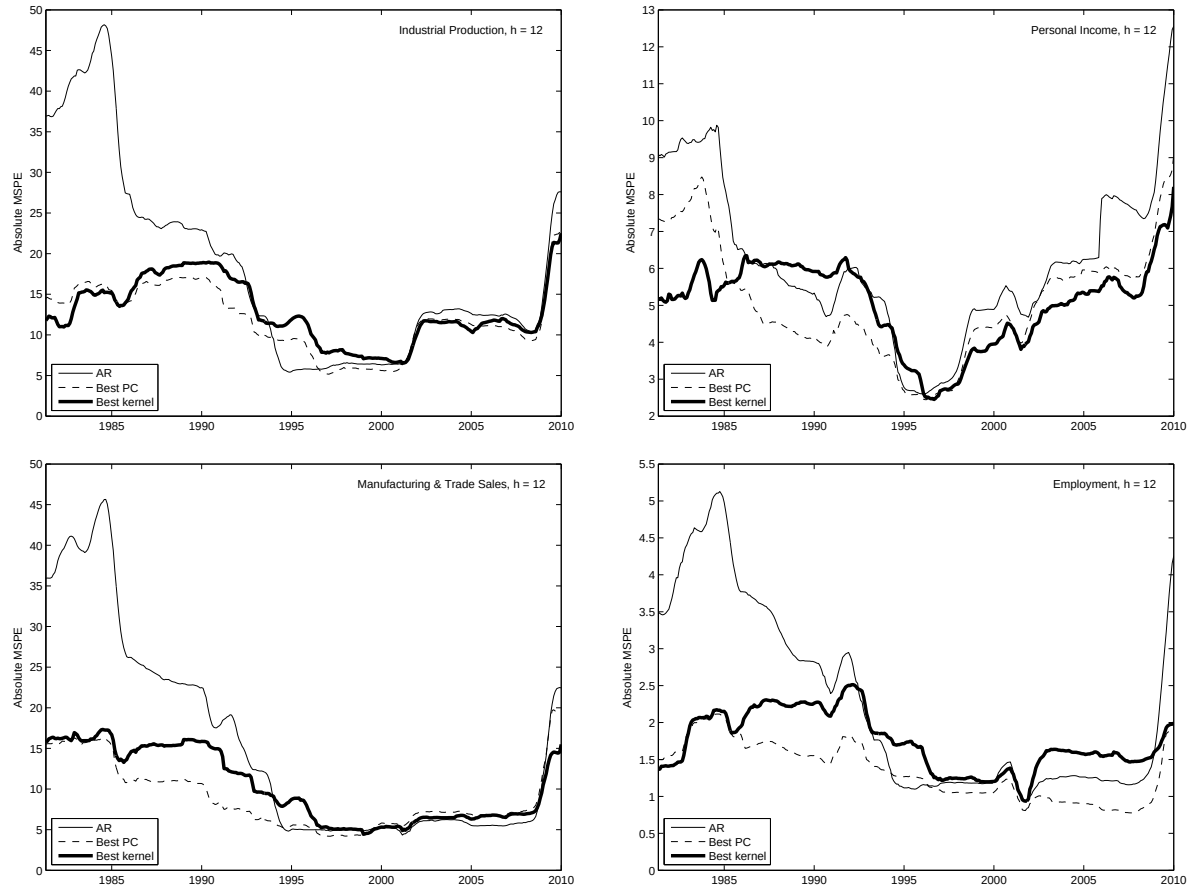
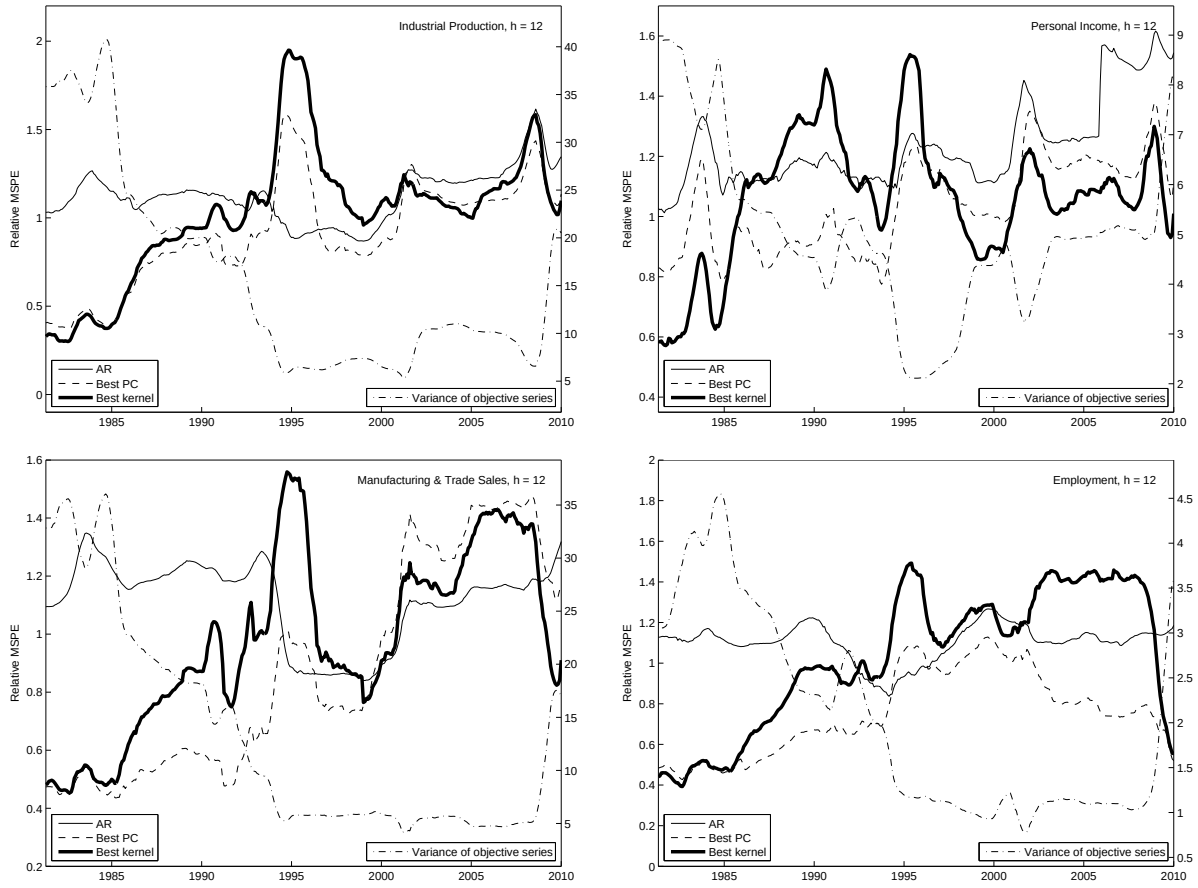


Figure 4 illustrates these conclusions by showing time series plots of the twelve-month growth rate of Personal Income. The choice of the three subperiods is motivated by dating the Great Moderation in 1984. The first subperiod contains only pre-Moderation data. As we estimate all models on 120-month rolling windows, the first forecast that is based only on post-Moderation data is the one for 1994, which marks the start of the last subperiod. During the second subperiod (see the middle panel of Figure 4), the kernel-based forecast is much more volatile than both the actual time series and the PC-based forecast. Apparently, kernel ridge regression is relatively more heavily affected by the break in volatility in the Personal Income series at the Great Moderation (with a variance of 7.84 for 1970-1983, 4.56 for 1984-1993, and 7.75 for 1984-2010). On both other subsamples, however, allowing for nonlinearity through kernel methods enhances the forecast quality considerably, see the top and bottom panels of Figure 4. The relative MSPEs, compared to the AR benchmark, for the three subperiods 1970-1983, 1984-1993, and 1994-2010 are respectively 86%, 71%, and 76% for PC, as compared to 72%, 87%, and 70% for

Figure 3: Ten-year rolling-window MSPEs for four macroeconomic series, for a forecast horizon of $h = 12$ months, for AR and for the best-performing PC and kernel methods, relative to the variance of the series being predicted.

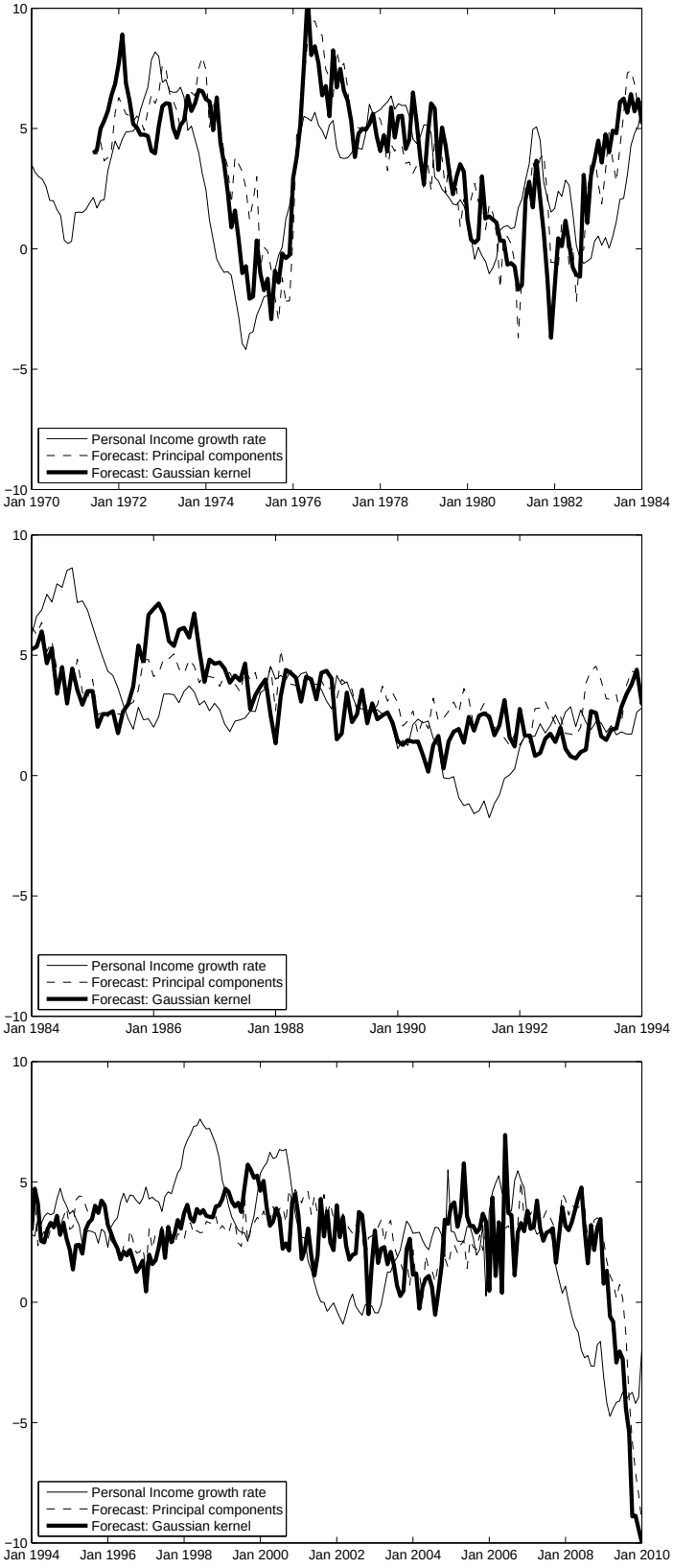


Gaussian kernel ridge regression. This result shows that the kernel method performs better than PC in the first and last subperiod. As a final point of interest, we note that the kernel-based method detects the 2008-9 crisis several months before the PC method does, as evidenced by the bottom panel of Figure 4.

5 Conclusion

We have introduced kernel ridge regression as a framework for accommodating nonlinear predictive relations in a data-rich environment. We have extended the existing kernel methodology to enable its use in time-series contexts typical for macroeconomic and financial applications. These extensions involve the incorporation of unpenalized linear terms in the forecast equation and an efficient leave-one-out cross-validation procedure for model selection. Our simulation study suggests that this method can deal with the type of data that comes up frequently in economic analysis, namely, data with a factor structure.

Figure 4: The twelve-month growth rate of Personal Income (thin line), with its PC-based forecast (dashed line) and its Gaussian-kernel forecast (heavy line). Top: 1970-1983. Middle: 1984-1993. Bottom: 1994-2010.



The empirical application to forecasting four key U.S. macroeconomic variables — production, income, sales, and employment — shows that kernel-based methods can provide more accurate forecasts than well-established autoregressive and principal-components-based methods. Further, kernel techniques consistently outperform previously proposed nonlinear extensions of the standard PC-based approach. Kernel ridge regression exhibits a relatively consistent good predictive performance, also during the crisis period in 2008-9. Among the kernel methods, the Gaussian kernel is found to produce the most reliable forecasts. This finding implies that it is not just the ridge term that contributes to the predictive accuracy, but accounting for nonlinearity leads to additional improvements in many cases. As using the Gaussian kernel does not require the forecaster to specify the form of nonlinearity in advance, this method is a powerful tool.

Finally, we have provided statistical evidence that kernel-based forecasts contain information that is missed by principal-components-based forecasts, and vice versa. This suggests a potential for forecast combinations. We conclude that the kernel methodology is a valuable addition to the macroeconomic forecaster's toolkit.

Appendix: Technical results

This appendix contains derivations of three results stated in Section 2. In Appendix A.1 we derive the expression for the forecast equation (3) for kernel ridge regression with additional unpenalized linear terms. In Appendix A.2 we obtain the mapping that corresponds to the Gaussian kernel function. Finally, in Appendix A.3 we describe an efficient leave-one-out cross-validation method for selecting tuning parameters in KRR.

A.1 Kernel ridge regression with unpenalized linear terms

We have shown in Section 2.2 that minimizing the penalized least-squares criterion $||y - Z\gamma||^2 + \lambda ||\gamma||^2$ leads to the forecast $\hat{y}_* = k_*' (K + \lambda I)^{-1} y$ as given in (2). In this appendix, we modify this forecast equation to allow for unpenalized linear terms as in the generalized forecast equation $\hat{y}_* = w_*'\hat{\beta} + z_*'\hat{\gamma}$, where the $P \times 1$ vector w_* contains the variables to be treated linearly. In this case, we seek to minimize

$$||y - W\beta - Z\gamma||^2 + \lambda ||\gamma||^2 \tag{A.1}$$

over the $P \times 1$ vector β and the $M \times 1$ vector γ. For given $\hat{\beta}$, we can proceed as in Section 2.2 to find

$$\hat{\gamma} = Z' (K + \lambda I)^{-1} \left(y - W\hat{\beta} \right). \tag{A.2}$$

On the other hand, for given $\hat{\gamma}$, minimizing criterion (A.1) is equivalent to ordinary least squares regression, which gives

$$\hat{\beta} = \left(W'W \right)^{-1} W' (y - Z\hat{\gamma}). \tag{A.3}$$

If we pre-multiply both sides of (A.3) by $W'W$, substitute the expression for $\hat{\gamma}$ from (A.2) into (A.3), and recall that $K = ZZ'$, we get

$$\begin{aligned} W'W\hat{\beta} &= W' \left(y - K (K + \lambda I)^{-1} \left(y - W\hat{\beta} \right) \right) \\ &= W'y - W'K (K + \lambda I)^{-1} y + W'K (K + \lambda I)^{-1} W\hat{\beta}. \end{aligned}$$

Collecting the terms involving $\hat{\beta}$ on the left-hand side of this equation, and rearranging, we obtain

$$W' \left(I - K (K + \lambda I)^{-1} \right) W\hat{\beta} = W' \left(I - K (K + \lambda I)^{-1} \right) y.$$

Using the fact that $I - K (K + \lambda I)^{-1} = (K + \lambda I - K) (K + \lambda I)^{-1} = \lambda (K + \lambda I)^{-1}$, this leads to the expression

$$\hat{\beta} = \left(W' (K + \lambda I)^{-1} W \right)^{-1} W' (K + \lambda I)^{-1} y.$$

If we then substitute this result and (A.2) into the forecast equation $\hat{y}_* = z_*'\hat{\gamma} + w_*'\hat{\beta}$, and recall that $k_* = Zz_*$, we find

$$\begin{aligned} \hat{y}_* &= k_*' (K + \lambda I)^{-1} \left(I - W \left(W' (K + \lambda I)^{-1} W \right)^{-1} W' (K + \lambda I)^{-1} \right) y \\ &\quad + w_*' \left(W' (K + \lambda I)^{-1} W \right)^{-1} W' (K + \lambda I)^{-1} y. \end{aligned} \tag{A.4}$$

To obtain a more manageable equation, note that by the partitioned matrix inversion formula,

$$\begin{pmatrix} K + \lambda I & W \\ W' & 0 \end{pmatrix}^{-1} = \begin{pmatrix} (K + \lambda I)^{-1} \left(I - WSW' (K + \lambda I)^{-1} \right) & (K + \lambda I)^{-1} WS \\ SW' (K + \lambda I)^{-1} & -S \end{pmatrix}, \tag{A.5}$$

where $S = \left(W' (K + \lambda I)^{-1} W \right)^{-1}$. It follows from this result that (A.4) is equivalent to the forecast equation (3) in Section 2.2:

$$\hat{y}_* = \begin{pmatrix} k_* \\ w_* \end{pmatrix}' \begin{pmatrix} K + \lambda I & W \\ W' & 0 \end{pmatrix}^{-1} \begin{pmatrix} y \\ 0 \end{pmatrix}.$$

A.2 Expansion of the Gaussian kernel

In this appendix, we derive the mapping φ that corresponds to the Gaussian kernel function. As stated in (5) in Section 2.3, this kernel function is defined as $\kappa(a,b) = \exp\left(||a-b||^2/2\right)$. If we write $-(1/2)||a-b||^2 = -a'a/2 - b'b/2 + a'b$ and expand the Taylor series for $\exp(a'b)$, we obtain

$$\kappa(a,b) = e^{-a'a/2}\, e^{-b'b/2} \sum_{r=0}^{\infty} \frac{1}{r!} (a'b)^r . \tag{A.6}$$

We proceed by expanding $(a'b)^r$ as a multinomial series:

$$(a'b)^r = \left(\sum_{n=1}^{N} a_n b_n\right)^r = \sum\sum\cdots\sum_{\left\{\sum_{n=1}^{N} d_n = r,\ \text{all } d_n \geq 0\right\}} \left(\frac{r!}{\prod_{n=1}^{N} d_n!} \prod_{n=1}^{N} (a_n b_n)^{d_n}\right).$$

Substituting this result into (A.6), we find

$$\begin{aligned}
\kappa(a,b) &= e^{-a'a/2}\, e^{-b'b/2} \sum_{r=0}^{\infty} \left(\frac{1}{r!} \sum\sum\cdots\sum_{\left\{\sum_{n=1}^{N} d_n = r,\ \text{all } d_n \geq 0\right\}} \left(\frac{r!}{\prod_{n=1}^{N} d_n!} \prod_{n=1}^{N} (a_n b_n)^{d_n}\right)\right) \\
&= e^{-a'a/2}\, e^{-b'b/2} \sum_{r=0}^{\infty} \left(\sum\sum\cdots\sum_{\left\{\sum_{n=1}^{N} d_n = r,\ \text{all } d_n \geq 0\right\}} \left(\prod_{n=1}^{N} \frac{(a_n b_n)^{d_n}}{d_n!}\right)\right) \\
&= e^{-a'a/2}\, e^{-b'b/2} \sum\sum\cdots\sum_{\{\text{all } d_n \geq 0,\ \text{for } n=1,2,\ldots,N\}} \left(\prod_{n=1}^{N} \frac{(a_n b_n)^{d_n}}{d_n!}\right).
\end{aligned}$$

Finally, we split the product into two factors that depend only on a and only on b, respectively:

$$\kappa(a,b) = \sum_{d_1=0}^{\infty} \sum_{d_2=0}^{\infty} \cdots \sum_{d_N=0}^{\infty} \left(e^{-a'a/2} \prod_{n=1}^{N} \frac{a_n^{d_n}}{\sqrt{d_n!}}\right) \left(e^{-b'b/2} \prod_{n=1}^{N} \frac{b_n^{d_n}}{\sqrt{d_n!}}\right). \tag{A.7}$$

As this expression shows, $\kappa(a,b) = \varphi(a)'\varphi(b)$, where, as claimed in Section 2.3, $\varphi(a)$ contains as elements, for each combination of degrees $d_1, d_2, \ldots, d_N \geq 0$,

$$e^{-a'a/2} \prod_{n=1}^{N} \frac{a_n^{d_n}}{\sqrt{d_n!}}.$$

A.3 Computationally efficient leave-one-out cross-validation

In this appendix, we describe an efficient method for leave-one-out cross-validation, which we employ to select the tuning parameters in KRR. Our derivation extends the results in Cawley and Talbot (2008) to allow for the unpenalized linear terms in the forecast equation (3). The result of Appendix A.1 can be summarized as follows: kernel ridge regression leads to the forecast

$$\hat{y}_* = \begin{pmatrix} k_* \\ w_* \end{pmatrix}' \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} \quad \text{with} \quad \begin{pmatrix} K + \lambda I & W \\ W' & 0 \end{pmatrix} \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix}. \tag{A.8}$$

The first step in leave-one-out cross-validation is to estimate the model on all observations except the first. As $K = ZZ'$, and each row of Z depends only on the corresponding row of X, the only elements in K that depend on the first observation are those in the first row and those in the first column. We therefore separate the first row and column from the other elements of K, and likewise, we split off the first row of W, and the first elements of $\hat{\alpha}$ and y. We denote these partitioned matrices and vectors by

$$K = \begin{pmatrix} k_{1,1} & k'_{-1,1} \\ k_{-1,1} & K_{-1,-1} \end{pmatrix}, \quad W = \begin{pmatrix} w'_1 \\ W_{-1} \end{pmatrix}, \quad \hat{\alpha} = \begin{pmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_{-1} \end{pmatrix} \quad \text{and} \quad y = \begin{pmatrix} y_1 \\ y_{-1} \end{pmatrix}.$$

We then have from (A.8)

$$\begin{pmatrix} k_{1,1} + \lambda & k'_{-1,1} & w'_1 \\ k_{-1,1} & K_{-1,-1} + \lambda I & W_{-1} \\ w_1 & W'_{-1} & 0 \end{pmatrix} \begin{pmatrix} \hat{\alpha}_1 \\ \hat{\alpha}_{-1} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_{-1} \\ 0 \end{pmatrix},$$

or equivalently, separating the first equation from the others,

$$\hat{\alpha}_1 (k_{1,1} + \lambda) + \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}' \begin{pmatrix} \hat{\alpha}_{-1} \\ \hat{\beta} \end{pmatrix} = y_1, \tag{A.9}$$

$$\hat{\alpha}_1 \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix} + \begin{pmatrix} K_{-1,1} + \lambda I & W_{-1} \\ W'_{-1} & 0 \end{pmatrix} \begin{pmatrix} \hat{\alpha}_{-1} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} y_{-1} \\ 0 \end{pmatrix}. \tag{A.10}$$

The forecast of y_1 based on a model estimated on observations $2, 3, \ldots, T$ clearly equals

$$\tilde{y}_1 = \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}' \begin{pmatrix} K_{-1,-1} + \lambda I & W_{-1} \\ W'_{-1} & 0 \end{pmatrix}^{-1} \begin{pmatrix} y_{-1} \\ 0 \end{pmatrix}$$

and using (A.9) and (A.10) we may write

$$\begin{aligned}
\tilde{y}_1 &= \hat{\alpha}_1 \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}' \begin{pmatrix} K_{-1,-1}+\lambda I & W_{-1} \\ W_{-1}' & 0 \end{pmatrix}^{-1} \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix} + \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}' \begin{pmatrix} \hat{\alpha}_{-1} \\ \hat{\beta} \end{pmatrix} \\
&= \hat{\alpha}_1 \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}' \begin{pmatrix} K_{-1,-1}+\lambda I & W_{-1} \\ W_{-1}' & 0 \end{pmatrix}^{-1} \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix} + y_1 - \hat{\alpha}_1 (k_{1,1}+\lambda) \\
&= y_1 - \hat{\alpha}_1 \left(k_{1,1}+\lambda - \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}' \begin{pmatrix} K_{-1,-1}+\lambda I & W_{-1} \\ W_{-1}' & 0 \end{pmatrix}^{-1} \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix} \right).
\end{aligned}$$

The expression $k_{1,1}+\lambda - \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}' \begin{pmatrix} K_{-1,-1}+\lambda I & W_{-1} \\ W_{-1}' & 0 \end{pmatrix}^{-1} \begin{pmatrix} k_{-1,1} \\ w_1 \end{pmatrix}$ is equal to the reciprocal of element $(1,1)$ of $\begin{pmatrix} k_{1,1}+\lambda & k_{-1,1}' & w_1' \\ k_{-1,1} & K_{-1,-1}+\lambda I & W_{-1} \\ w_1 & W_{-1}' & 0 \end{pmatrix}^{-1} = \begin{pmatrix} K+\lambda I & W \\ W' & 0 \end{pmatrix}^{-1}$, as can be seen by using the partitioned matrix inversion formula. Therefore, the first leave-one-out error equals

$$y_1 - \tilde{y}_1 = \hat{\alpha}_1 \Big/ \text{element } (1,1) \text{ of } \begin{pmatrix} K+\lambda I & W \\ W' & 0 \end{pmatrix}^{-1}.$$

In general, an analogous derivation shows that the t-th leave-one-out prediction error equals

$$y_t - \tilde{y}_t = \hat{\alpha}_t \Big/ \text{element } (t,t) \text{ of } \begin{pmatrix} K+\lambda I & W \\ W' & 0 \end{pmatrix}^{-1}. \tag{A.11}$$

That is, we obtain all leave-one-out errors by dividing each element of the vector $\hat{\alpha}$ by the corresponding diagonal element of the matrix $\begin{pmatrix} K+\lambda I & W \\ W' & 0 \end{pmatrix}^{-1}$. Observe that both $\hat{\alpha}$ and this inverse are needed in computing the forecast $\hat{y}_*$. Thus, in the process of making the out-of-sample prediction, we can find all leave-one-out errors without performing any additional computations, aside from the division in (A.11).

As a final note, we mention that the matrix inverse in (A.11) can also be computed efficiently. As $K+\lambda I$ is symmetric and positive definite, its inverse can be computed from its Cholesky decomposition. The inverse of the full matrix can then be calculated using (A.5) in Appendix A.1.

References

M. Aiolfi and C.A. Favero. Model uncertainty, thick modelling and the predictability of stock returns. *Journal of Forecasting*, 24:233–254, 2005.

J. Bai and S. Ng. Forecasting economic time series using targeted predictors. *Journal of Econometrics*, 146:304–317, 2008.

M. Bańbura, D. Giannone, and L. Reichlin. Large Bayesian vector autoregressions. *Journal of Applied Econometrics*, 25:71–92, 2010.

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.

A. Carriero, G. Kapetanios, and M. Marcellino. Forecasting large datasets with Bayesian reduced rank multivariate models. *Journal of Applied Econometrics*, 26:735–761, 2011.

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.

C. Çakmaklı and D. van Dijk. Getting the most out of macroeconomic information for predicting stock returns and volatility. *Tinbergen Institute Discussion Paper 2010-115/4*, 2010.

C. De Mol, D. Giannone, and L. Reichlin. Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146:318–328, 2008.

P. Exterkate. Model selection in kernel ridge regression. *CREATES Research Paper 2012-10*, 2012.

J. Faust and J.H. Wright. Comparing Greenbook and reduced form forecasts using a large realtime dataset. *Journal of Business and Economic Statistics*, 27:468–479, 2009.

B.C. Giovannetti. Nonlinear forecasting using factor-augmented models. *Journal of Forecasting*, forthcoming, doi:10.1002/for.1248, 2011.

J.J.J. Groen and G. Kapetanios. Revisiting useful approaches to data-rich macroeconomic forecasting. *Federal Reserve Bank of New York Staff Report 327*, 2008.

H. Huang and T.-H. Lee. To combine forecasts or to combine information? *Econometric Reviews*, 29: 534–570, 2010.

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.

K.-R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks ICANN'97*, pages 999–1004. Springer, Berlin, 1997.

A.R. Pagan and A. Ullah. *Nonparametric Econometrics*. Cambridge University Press, Cambridge, 1999.

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

J. Racine. Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics*, 99:39–61, 2000.

D.E. Rapach, J.K. Strauss, and G. Zhou. Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *Review of Financial Studies*, 23:821–862, 2010.

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 and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14: 199–222, 2004.

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.

J.H. Stock and M.W. Watson. Implications of dynamic factor models for VAR analysis. *NBER Working Paper No. 11467*, 2005.

J.H. Stock and M.W. Watson. Forecasting with many predictors. In G. Elliot, C.W.J. Granger, and A. Timmermann, editors, *Handbook of Economic Forecasting*, pages 515–554. Elsevier, Amsterdam, 2006.

J.H. Stock and M.W. Watson. Why has U.S. inflation become harder to forecast? *Journal of Money, Credit and Banking*, 39:3–33, 2007.

J.H. Stock and M.W. Watson. Generalized shrinkage methods for forecasting using many predictors. *Manuscript, Harvard University*, 2009.

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.

V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

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.

J.H. Wright. Forecasting US inflation by Bayesian model averaging. *Journal of Forecasting*, 28:131–144, 2009.