Nonlinear Forecasting with Many Predictors using Kernel Ridge Regression

Peter Exterkate

CREATES
Center for Research in Econometric Analysis of Time Series

AARHUS UNIVERSITY

Seventh ECB Workshop on Forecasting Techniques
New Directions for Forecasting

Frankfurt am Main, May 4, 2012

One-slide summary

One-slide summary

- **Main research question:** Is it possible to forecast with large data sets, while allowing for nonlinear relations between target variable and predictors?

One-slide summary

- **Main research question:** Is it possible to forecast with large data sets, while allowing for nonlinear relations between target variable and predictors?
- **Background:** Large data sets are increasingly available in macroeconomics and finance, but forecasting is mostly limited to a linear framework

One-slide summary

- **Main research question:** Is it possible to forecast with large data sets, while allowing for nonlinear relations between target variable and predictors?
- **Background:** Large data sets are increasingly available in macroeconomics and finance, but forecasting is mostly limited to a linear framework
- **Solution:** Kernel ridge regression (KRR), which avoids the curse of dimensionality by manipulating the forecast equation in a clever way: the *kernel trick*

One-slide summary

- **Main research question:** Is it possible to forecast with large data sets, while allowing for nonlinear relations between target variable and predictors?
- **Background:** Large data sets are increasingly available in macroeconomics and finance, but forecasting is mostly limited to a linear framework
- **Solution:** Kernel ridge regression (KRR), which avoids the curse of dimensionality by manipulating the forecast equation in a clever way: the *kernel trick*
- **Contributions:**
  - Extension of KRR to models with "preferred" predictors
  - Monte Carlo and empirical evidence that KRR works, and improves upon conventional techniques such as principal component regression
  - Clearer understanding of the choice of kernel and tuning parameters (companion paper)

One-slide summary

- **Main research question:** Is it possible to forecast with large data sets, while allowing for nonlinear relations between target variable and predictors?
- **Background:** Large data sets are increasingly available in macroeconomics and finance, but forecasting is mostly limited to a linear framework
- **Solution:** Kernel ridge regression (KRR), which avoids the curse of dimensionality by manipulating the forecast equation in a clever way: the *kernel trick*
- **Contributions:**
  - Extension of KRR to models with "preferred" predictors
  - Monte Carlo and empirical evidence that KRR works, and improves upon conventional techniques such as principal component regression
  - Clearer understanding of the choice of kernel and tuning parameters (companion paper)
- Joint work with Patrick Groenen, Christiaan Heij, and Dick van Dijk (Econometric Institute, Erasmus University Rotterdam)

Introduction

Introduction

- How to forecast in today's data-rich environment?

Introduction

- How to forecast in today's data-rich environment?
- In an ideal world:
  - use all available information
  - flexible functional forms

Introduction

- How to forecast in today's data-rich environment?
- In an ideal world:
  - use all available information
  - flexible functional forms
- In practice:
  - the simpler the better
  - "curse of dimensionality"

Possible ways out

Possible ways out

- Handling high-dimensionality:

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)
  - Selecting variables (Bai and Ng, 2008)

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)
  - Selecting variables (Bai and Ng, 2008)
  - Bayesian regression (De Mol, Giannone, Reichlin, 2008)

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)
  - Selecting variables (Bai and Ng, 2008)
  - Bayesian regression (De Mol, Giannone, Reichlin, 2008)

- Handling nonlinearity:

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)
  - Selecting variables (Bai and Ng, 2008)
  - Bayesian regression (De Mol, Giannone, Reichlin, 2008)
- Handling nonlinearity:
  - Neural networks (Teräsvirta, Van Dijk, Medeiros, 2005)

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)
  - Selecting variables (Bai and Ng, 2008)
  - Bayesian regression (De Mol, Giannone, Reichlin, 2008)
- Handling nonlinearity:
  - Neural networks (Teräsvirta, Van Dijk, Medeiros, 2005)
  - Linear regression on nonlinear PCs (Bai and Ng, 2008)

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)
  - Selecting variables (Bai and Ng, 2008)
  - Bayesian regression (De Mol, Giannone, Reichlin, 2008)
- Handling nonlinearity:
  - Neural networks (Teräsvirta, Van Dijk, Medeiros, 2005)
  - Linear regression on nonlinear PCs (Bai and Ng, 2008)
  - Nonlinear regression on linear PCs (Giovannetti, 2011)

Possible ways out

- Handling high-dimensionality:
  - Principal components regression (Stock and Watson, 2002)
  - Partial least squares (Groen and Kapetanios, 2008)
  - Selecting variables (Bai and Ng, 2008)
  - Bayesian regression (De Mol, Giannone, Reichlin, 2008)
- Handling nonlinearity:
  - Neural networks (Teräsvirta, Van Dijk, Medeiros, 2005)
  - Linear regression on nonlinear PCs (Bai and Ng, 2008)
  - Nonlinear regression on linear PCs (Giovannetti, 2011)
- Unified approach: kernel ridge regression

Forecasting context

Forecasting context

- We aim to forecast $y_* \in \mathbb{R}$, using a set of predictors $x_* \in \mathbb{R}^N$

Forecasting context

- We aim to forecast $y_* \in \mathbb{R}$, using a set of predictors $x_* \in \mathbb{R}^N$
- Historical observations are collected in $y \in \mathbb{R}^T$ and $X \in \mathbb{R}^{T \times N}$

Forecasting context

- We aim to forecast $y_* \in \mathbb{R}$, using a set of predictors $x_* \in \mathbb{R}^N$
- Historical observations are collected in $y \in \mathbb{R}^T$ and $X \in \mathbb{R}^{T \times N}$
- Assuming a linear relation, we would use OLS to minimize $||y - X\beta||^2$

Forecasting context

- We aim to forecast $y_* \in \mathbb{R}$, using a set of predictors $x_* \in \mathbb{R}^N$
- Historical observations are collected in $y \in \mathbb{R}^T$ and $X \in \mathbb{R}^{T \times N}$
- Assuming a linear relation, we would use OLS to minimize $||y - X\beta||^2$
- Forecast would be $\hat{y}_* = x_*'\hat{\beta} = x_*'(X'X)^{-1}X'y$

Forecasting context

- We aim to forecast $y_* \in \mathbb{R}$, using a set of predictors $x_* \in \mathbb{R}^N$
- Historical observations are collected in $y \in \mathbb{R}^T$ and $X \in \mathbb{R}^{T \times N}$
- Assuming a linear relation, we would use OLS to minimize $||y - X\beta||^2$
- Forecast would be $\hat{y}_* = x_*' \hat{\beta} = x_*' \left(X'X\right)^{-1} X'y$
- This requires $N \leq T$ (in theory) or $N \ll T$ (in practice)

Ridge regression

Ridge regression

- A standard solution is ridge regression: given some $\lambda > 0$, minimize $||y - X\beta||^2 + \lambda\, ||\beta||^2$

Ridge regression

- A standard solution is ridge regression: given some $\lambda > 0$, minimize $||y - X\beta||^2 + \lambda\,||\beta||^2$
- In this case, the forecast becomes $\hat{y}_* = x_*'\hat{\beta} = x_*'\left(X'X + \lambda I\right)^{-1} X'y$, even if $N > T$

Ridge regression

- A standard solution is ridge regression: given some $\lambda > 0$, minimize $||y - X\beta||^2 + \lambda ||\beta||^2$
- In this case, the forecast becomes $\hat{y}_* = x_*' \hat{\beta} = x_*' (X'X + \lambda I)^{-1} X'y$, even if $N > T$
- So, for nonlinear forecasts, let $z = \varphi(x)$ with $\varphi : \mathbb{R}^N \to \mathbb{R}^M$, and $\hat{y}_* = z_*' (Z'Z + \lambda I)^{-1} Z'y$

Ridge regression

- A standard solution is ridge regression: given some $\lambda > 0$, minimize $||y - X\beta||^2 + \lambda ||\beta||^2$
- In this case, the forecast becomes $\hat{y}_* = x_*' \hat{\beta} = x_*' (X'X + \lambda I)^{-1} X'y$, even if $N > T$
- So, for nonlinear forecasts, let $z = \varphi(x)$ with $\varphi : \mathbb{R}^N \to \mathbb{R}^M$, and $\hat{y}_* = z_*' (Z'Z + \lambda I)^{-1} Z'y$
- For very large M, the inversion is numerically unstable and computationally intensive

Ridge regression

- A standard solution is ridge regression: given some $\lambda > 0$, minimize $||y - X\beta||^2 + \lambda ||\beta||^2$
- In this case, the forecast becomes $\hat{y}_* = x_*' \hat{\beta} = x_*' (X'X + \lambda I)^{-1} X'y$, even if $N > T$
- So, for nonlinear forecasts, let $z = \varphi(x)$ with $\varphi : \mathbb{R}^N \to \mathbb{R}^M$, and $\hat{y}_* = z_*' (Z'Z + \lambda I)^{-1} Z'y$
- For very large M, the inversion is numerically unstable and computationally intensive
- Typical example: $N = 132$, quadratic model $\Rightarrow M = 8911$

Kernel trick (Boser, Guyon, Vapnik, 1992)

Kernel trick (Boser, Guyon, Vapnik, 1992)

- Essential idea: if $M \gg T$, working with T-dimensional objects is easier than working with M-dimensional objects

Kernel trick (Boser, Guyon, Vapnik, 1992)

- Essential idea: if $M \gg T$, working with T-dimensional objects is easier than working with M-dimensional objects
- We wish to compute $\hat{y}_* = z_*' \left(Z'Z + \lambda I\right)^{-1} Z'y$

Kernel trick (Boser, Guyon, Vapnik, 1992)

- Essential idea: if $M \gg T$, working with T-dimensional objects is easier than working with M-dimensional objects
- We wish to compute $\hat{y}_* = z_*' \left(Z'Z + \lambda I\right)^{-1} Z'y$
- Some algebra yields $\hat{y}_* = z_*' Z' \left(ZZ' + \lambda I\right)^{-1} y$

Kernel trick (Boser, Guyon, Vapnik, 1992)

- Essential idea: if $M \gg T$, working with T-dimensional objects is easier than working with M-dimensional objects
- We wish to compute $\hat{y}_* = z_*' \left(Z'Z + \lambda I\right)^{-1} Z'y$
- Some algebra yields $\hat{y}_* = z_*' Z' \left(ZZ' + \lambda I\right)^{-1} y$
- So if we know $k_* = Zz_* \in \mathbb{R}^T$ and $K = ZZ' \in \mathbb{R}^{T \times T}$, computing $\hat{y}_* = k_*' \left(K + \lambda I\right)^{-1} y$ is feasible

Kernel trick (Boser, Guyon, Vapnik, 1992)

- Essential idea: if $M \gg T$, working with T-dimensional objects is easier than working with M-dimensional objects
- We wish to compute $\hat{y}_* = z_*' \left(Z'Z + \lambda I\right)^{-1} Z'y$
- Some algebra yields $\hat{y}_* = z_*' Z' \left(ZZ' + \lambda I\right)^{-1} y$
- So if we know $k_* = Zz_* \in \mathbb{R}^T$ and $K = ZZ' \in \mathbb{R}^{T \times T}$, computing $\hat{y}_* = k_*' \left(K + \lambda I\right)^{-1} y$ is feasible

- Define the kernel function $\kappa\left(x_s, x_t\right) = \varphi\left(x_s\right)' \varphi\left(x_t\right)$
  - tth element of k_* is $z_t' z_* = \kappa\left(x_t, x_*\right)$
  - (s, t)th element of K is $z_s' z_t = \kappa\left(x_s, x_t\right)$

Kernel trick (Boser, Guyon, Vapnik, 1992)

- Essential idea: if $M \gg T$, working with T-dimensional objects is easier than working with M-dimensional objects
- We wish to compute $\hat{y}_* = z_*' \left(Z'Z + \lambda I\right)^{-1} Z'y$
- Some algebra yields $\hat{y}_* = z_*' Z' \left(ZZ' + \lambda I\right)^{-1} y$
- So if we know $k_* = Zz_* \in \mathbb{R}^T$ and $K = ZZ' \in \mathbb{R}^{T \times T}$, computing $\hat{y}_* = k_*' \left(K + \lambda I\right)^{-1} y$ is feasible

- Define the kernel function $\kappa\left(x_s, x_t\right) = \varphi\left(x_s\right)' \varphi\left(x_t\right)$
  - tth element of k_* is $z_t' z_* = \kappa\left(x_t, x_*\right)$
  - (s, t)th element of K is $z_s' z_t = \kappa\left(x_s, x_t\right)$
- If we choose φ smartly, κ (and hence $\hat{y}_*$) will be easy to compute!

Bayesian interpretation

Bayesian interpretation

- Like “normal” ridge regression, KRR has a Bayesian interpretation:

Bayesian interpretation

- Like “normal” ridge regression, KRR has a Bayesian interpretation:
- Likelihood: $p\left(y|X,\beta,\theta^2\right)=\mathcal{N}\left(Z\beta,\theta^2 I\right)$

Bayesian interpretation

- Like "normal" ridge regression, KRR has a Bayesian interpretation:
- Likelihood: $p\left(y|X,\beta,\theta^2\right) = \mathcal{N}\left(Z\beta,\theta^2 I\right)$
- Priors: $p\left(\theta^2\right) \propto \theta^{-2}$, $p\left(\beta|\theta\right) = \mathcal{N}\left(0,\left(\theta^2/\lambda\right) I\right)$

Bayesian interpretation

- Like "normal" ridge regression, KRR has a Bayesian interpretation:
- Likelihood: $p\left(y|X,\beta,\theta^2\right) = \mathcal{N}\left(Z\beta,\theta^2 I\right)$
- Priors: $p\left(\theta^2\right) \propto \theta^{-2}$, $p\left(\beta|\theta\right) = \mathcal{N}\left(0,\left(\theta^2/\lambda\right) I\right)$
- Posterior distribution of y_* is Student's t with T degrees of freedom, mode $\hat{y}_*$, variance also analytically available

Bayesian interpretation

- Like "normal" ridge regression, KRR has a Bayesian interpretation:
- Likelihood: $p\left(y|X, \beta, \theta^2\right) = \mathcal{N}\left(Z\beta, \theta^2 I\right)$
- Priors: $p\left(\theta^2\right) \propto \theta^{-2}$, $p\left(\beta|\theta\right) = \mathcal{N}\left(0, \left(\theta^2/\lambda\right) I\right)$
- Posterior distribution of y_* is Student's t with T degrees of freedom, mode $\hat{y}_*$, variance also analytically available
- Note that we can interpret λ in terms of the signal-to-noise ratio

Function approximation (Hofmann, Schölkopf, Smola, 2008)

Function approximation (Hofmann, Schölkopf, Smola, 2008)

- Other way to look at KRR: it also solves, for some Hilbert space $\mathcal{H}$,

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

Function approximation (Hofmann, Schölkopf, Smola, 2008)

- Other way to look at KRR: it also solves, for some Hilbert space $\mathcal{H}$,

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

- Choosing a kernel function implies choosing $\mathcal{H}$ and its norm $||\cdot||_{\mathcal{H}}$

Function approximation (Hofmann, Schölkopf, Smola, 2008)

- Other way to look at KRR: it also solves, for some Hilbert space $\mathcal{H}$,

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

- Choosing a kernel function implies choosing $\mathcal{H}$ and its norm $\left|\left|\cdot\right|\right|_{\mathcal{H}}$
- The "complexity" of the prediction function is measured by $\left|\left|f\right|\right|_{\mathcal{H}}$

Choosing the kernel function

- We can understand KRR from a Bayesian/ridge point of view, or as a function approximation technique

Choosing the kernel function

- We can understand KRR from a Bayesian/ridge point of view, or as a function approximation technique
- Thus, our choice of kernel can be guided in two ways:

Choosing the kernel function

- We can understand KRR from a Bayesian/ridge point of view, or as a function approximation technique
- Thus, our choice of kernel can be guided in two ways:
  - The prediction function $x \mapsto y$ will be linear in $\varphi(x)$, so choose a κ that leads to a φ for which this makes sense

Choosing the kernel function

- We can understand KRR from a Bayesian/ridge point of view, or as a function approximation technique
- Thus, our choice of kernel can be guided in two ways:
  - The prediction function $x \mapsto y$ will be linear in $\varphi(x)$, so choose a κ that leads to a φ for which this makes sense
  - Complexity of the prediction function is penalized through $||\cdot||_{\mathcal{H}}$, so choose a κ for which this penalty ensures "smoothness"

Choosing the kernel function

- We can understand KRR from a Bayesian/ridge point of view, or as a function approximation technique
- Thus, our choice of kernel can be guided in two ways:
  - The prediction function $x \mapsto y$ will be linear in $\varphi(x)$, so choose a κ that leads to a φ for which this makes sense
  - Complexity of the prediction function is penalized through $||\cdot||_{\mathcal{H}}$, so choose a κ for which this penalty ensures "smoothness"
- We will give examples of both

Polynomial kernel functions (Poggio, 1975)

Polynomial kernel functions (Poggio, 1975)

- Linear ridge regression: $\varphi(x) = x$ implies $\kappa(x_s, x_t) = x_s' x_t$

Polynomial kernel functions (Poggio, 1975)

- Linear ridge regression: $\varphi(x) = x$ implies $\kappa(x_s, x_t) = x_s' x_t$
- Obvious extension: $\varphi(x) = \left(1, x_1, x_2, \ldots, x_1^2, x_2^2, \ldots, x_1 x_2, \ldots\right)'$

Polynomial kernel functions (Poggio, 1975)

- Linear ridge regression: $\varphi(x) = x$ implies $\kappa(x_s, x_t) = x_s' x_t$
- Obvious extension: $\varphi(x) = \left(1, x_1, x_2, \ldots, x_1^2, x_2^2, \ldots, x_1 x_2, \ldots\right)'$
- However, κ does not take a particularly simple form in this case

Polynomial kernel functions (Poggio, 1975)

- Linear ridge regression: $\varphi(x) = x$ implies $\kappa(x_s, x_t) = x_s' x_t$
- Obvious extension: $\varphi(x) = \left(1, x_1, x_2, \ldots, x_1^2, x_2^2, \ldots, x_1 x_2, \ldots\right)'$
- However, κ does not take a particularly simple form in this case
- Better: $\varphi(x) = \left(1, \frac{\sqrt{2}}{\sigma} x_1, \frac{\sqrt{2}}{\sigma} x_2, \ldots, \frac{1}{\sigma^2} x_1^2, \frac{1}{\sigma^2} x_2^2, \ldots, \frac{\sqrt{2}}{\sigma^2} x_1 x_2, \ldots\right)'$,
  which implies $\kappa(x_s, x_t) = \left(1 + \frac{x_s' x_t}{\sigma^2}\right)^2$

Polynomial kernel functions (Poggio, 1975)

- Linear ridge regression: $\varphi(x) = x$ implies $\kappa(x_s, x_t) = x_s' x_t$
- Obvious extension: $\varphi(x) = \left(1, x_1, x_2, \ldots, x_1^2, x_2^2, \ldots, x_1 x_2, \ldots\right)'$
- However, κ does not take a particularly simple form in this case
- Better: $\varphi(x) = \left(1, \frac{\sqrt{2}}{\sigma} x_1, \frac{\sqrt{2}}{\sigma} x_2, \ldots, \frac{1}{\sigma^2} x_1^2, \frac{1}{\sigma^2} x_2^2, \ldots, \frac{\sqrt{2}}{\sigma^2} x_1 x_2, \ldots\right)'$, which implies $\kappa(x_s, x_t) = \left(1 + \frac{x_s' x_t}{\sigma^2}\right)^2$
- More generally, $\kappa(x_s, x_t) = \left(1 + \frac{x_s' x_t}{\sigma^2}\right)^d$ corresponds to $\varphi(x) =$ (all monomials in x up to degree d)

Polynomial kernel functions (Poggio, 1975)

- Linear ridge regression: $\varphi(x) = x$ implies $\kappa(x_s, x_t) = x_s' x_t$
- Obvious extension: $\varphi(x) = \left(1, x_1, x_2, \ldots, x_1^2, x_2^2, \ldots, x_1 x_2, \ldots\right)'$
- However, κ does not take a particularly simple form in this case
- Better: $\varphi(x) = \left(1, \frac{\sqrt{2}}{\sigma} x_1, \frac{\sqrt{2}}{\sigma} x_2, \ldots, \frac{1}{\sigma^2} x_1^2, \frac{1}{\sigma^2} x_2^2, \ldots, \frac{\sqrt{2}}{\sigma^2} x_1 x_2, \ldots\right)'$, which implies $\kappa(x_s, x_t) = \left(1 + \frac{x_s' x_t}{\sigma^2}\right)^2$
- More generally, $\kappa(x_s, x_t) = \left(1 + \frac{x_s' x_t}{\sigma^2}\right)^d$ corresponds to $\varphi(x) =$ (all monomials in x up to degree d)
- Interpretation of tuning parameter: higher $\sigma \Rightarrow$ smaller coefficients on higher-order terms $\Rightarrow$ smoother prediction function

The Gaussian kernel function (Broomhead and Lowe, 1988)

The Gaussian kernel function (Broomhead and Lowe, 1988)

- Examine the effects of $||f||_{\mathcal{H}}$ on $\tilde{f}$, the Fourier transform of the prediction function. Popular choice: set the kernel κ such that

$$||f||_{\mathcal{H}} \propto \int_{\mathbb{R}^N} \frac{\left|\tilde{f}\left(\omega\right)\right|^2}{\sigma^N \exp\left(-\frac{1}{2}\sigma^2\omega'\omega\right)} d\omega$$

The Gaussian kernel function (Broomhead and Lowe, 1988)

- Examine the effects of $||f||_{\mathcal{H}}$ on $\tilde{f}$, the Fourier transform of the prediction function. Popular choice: set the kernel κ such that

$$||f||_{\mathcal{H}} \propto \int_{\mathbb{R}^N} \frac{\left|\tilde{f}(\omega)\right|^2}{\sigma^N \exp\left(-\frac{1}{2}\sigma^2\omega'\omega\right)} d\omega$$

- As $\sigma \uparrow$, components at high frequencies ω are penalized more heavily, leading to a smoother f

The Gaussian kernel function (Broomhead and Lowe, 1988)

- Examine the effects of $||f||_{\mathcal{H}}$ on $\tilde{f}$, the Fourier transform of the prediction function. Popular choice: set the kernel κ such that

$$||f||_{\mathcal{H}} \propto \int_{\mathbb{R}^N} \frac{\left|\tilde{f}(\omega)\right|^2}{\sigma^N \exp\left(-\frac{1}{2}\sigma^2\omega'\omega\right)} d\omega$$

- As $\sigma \uparrow$, components at high frequencies ω are penalized more heavily, leading to a smoother f

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

The Gaussian kernel function (Broomhead and Lowe, 1988)

- Examine the effects of $||f||_{\mathcal{H}}$ on $\tilde{f}$, the Fourier transform of the prediction function. Popular choice: set the kernel κ such that

$$||f||_{\mathcal{H}} \propto \int_{\mathbb{R}^N} \frac{\left|\tilde{f}(\omega)\right|^2}{\sigma^N \exp\left(-\frac{1}{2}\sigma^2\omega'\omega\right)} d\omega$$

- As $\sigma \uparrow$, components at high frequencies ω are penalized more heavily, leading to a smoother f
- Corresponding kernel is $\kappa(x_s, x_t) = \exp\left(\frac{-1}{2\sigma^2}||x_s - x_t||^2\right)$
- For a ridge regression interpretation, we would need to build *infinitely many* regressors of the form $\exp\left(-\frac{x'x}{2\sigma^2}\right)\prod_{n=1}^{N}\frac{x_n^{d_n}}{\sigma^{d_n}\sqrt{d_n!}}$, for nonnegative integers $d_1, d_2, \ldots, d_N$. Thus, the kernel trick allows us to implicitly work with an infinite number of regressors

Tuning parameters

Tuning parameters

- Several tuning parameters:

Tuning parameters

- Several tuning parameters:
  - Penalty parameter λ
  - Smoothness parameter σ

Tuning parameters

- Several tuning parameters:
  - Penalty parameter λ
  - Smoothness parameter σ
  - In our application: lag lengths (for y and X)

Tuning parameters

- Several tuning parameters:
  - Penalty parameter λ
  - Smoothness parameter σ
  - In our application: lag lengths (for y and X)

- Leave-one-out cross-validation can be implemented in a computationally efficient way (Cawley and Talbot, 2008)

Tuning parameters

- Several tuning parameters:
  - Penalty parameter λ
  - Smoothness parameter σ
  - In our application: lag lengths (for y and X)

- Leave-one-out cross-validation can be implemented in a computationally efficient way (Cawley and Talbot, 2008)

- A small (5×5) grid of "reasonable" values for λ and σ is proposed in a companion paper (Exterkate, February 2012)

"Preferred" predictors

“Preferred” predictors

- In econometrics, we often want to include some “preferred” predictors (e.g. lags of y) individually, linearly, and without penalizing their coefficients

"Preferred" predictors

- In econometrics, we often want to include some "preferred" predictors (e.g. lags of y) individually, linearly, and without penalizing their coefficients

- Thus, instead of $y_t = \varphi(x_t)' \beta + u_t$, we aim to estimate $y_t = w_t' \gamma + \varphi(x_t)' \beta + u_t$

"Preferred" predictors

- In econometrics, we often want to include some "preferred" predictors (e.g. lags of y) individually, linearly, and without penalizing their coefficients

- Thus, instead of $y_t = \varphi(x_t)' \beta + u_t$, we aim to estimate $y_t = w_t' \gamma + \varphi(x_t)' \beta + u_t$

- We show that replacing $\hat{y}_* = k_*' (K + \lambda I)^{-1} y$ by $\hat{y}_* = \begin{pmatrix} k_* \\ w_* \end{pmatrix}' \begin{pmatrix} K + \lambda I & W \\ W' & 0 \end{pmatrix}^{-1} \begin{pmatrix} y \\ 0 \end{pmatrix}$ solves this problem

"Preferred" predictors

- In econometrics, we often want to include some "preferred" predictors (e.g. lags of y) individually, linearly, and without penalizing their coefficients
- Thus, instead of $y_t = \varphi(x_t)' \beta + u_t$, we aim to estimate $y_t = w_t' \gamma + \varphi(x_t)' \beta + u_t$
- We show that replacing $\hat{y}_* = k_*' (K + \lambda I)^{-1} y$ by $\hat{y}_* = \begin{pmatrix} k_* \\ w_* \end{pmatrix}' \begin{pmatrix} K + \lambda I & W \\ W' & 0 \end{pmatrix}^{-1} \begin{pmatrix} y \\ 0 \end{pmatrix}$ solves this problem
- Computationally efficient leave-one-out cross-validation still works

Time-series models

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \ldots, y_{t-p+1}$?

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \ldots, y_{t-p+1}$?
  - Recall Bayesian interpretation and write $p(y) = p(y_1, \ldots, y_p) \cdot p(y_{p+1}|y_p, \ldots, y_1) \cdots p(y_T|y_{T-1}, \ldots, y_1)$

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \dots, y_{t-p+1}$?
  - Recall Bayesian interpretation and write $p(y) = p(y_1, \dots, y_p) \cdot p(y_{p+1} | y_p, \dots, y_1) \cdots p(y_T | y_{T-1}, \dots, y_1)$
  - Nothing changes, provided that we condition on p initial values

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \ldots, y_{t-p+1}$?
  - Recall Bayesian interpretation and write $p(y) = p(y_1, \ldots, y_p) \cdot p(y_{p+1}|y_p, \ldots, y_1) \cdots p(y_T|y_{T-1}, \ldots, y_1)$
  - Nothing changes, provided that we condition on p initial values
  - Even stationarity does not seem to be an issue

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \ldots, y_{t-p+1}$?
  - Recall Bayesian interpretation and write $p(y) = p(y_1, \ldots, y_p) \cdot p(y_{p+1}|y_p, \ldots, y_1) \cdots p(y_T|y_{T-1}, \ldots, y_1)$
  - Nothing changes, provided that we condition on p initial values
  - Even stationarity does not seem to be an issue
- What if y_t is multivariate?

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \ldots, y_{t-p+1}$?
  - Recall Bayesian interpretation and write $p(y) = p(y_1, \ldots, y_p) \cdot p(y_{p+1}|y_p, \ldots, y_1) \cdots p(y_T|y_{T-1}, \ldots, y_1)$
  - Nothing changes, provided that we condition on p initial values
  - Even stationarity does not seem to be an issue
- What if y_t is multivariate?
  - No problem whatsoever, whether or not $E_{t-1}[u_t u_t']$ is diagonal
  - So, we could treat e.g. nonlinear VAR-like models

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \ldots, y_{t-p+1}$?
  - Recall Bayesian interpretation and write $p(y) = p(y_1, \ldots, y_p) \cdot p(y_{p+1}|y_p, \ldots, y_1) \cdots p(y_T|y_{T-1}, \ldots, y_1)$
  - Nothing changes, provided that we condition on p initial values
  - Even stationarity does not seem to be an issue
- What if y_t is multivariate?
  - No problem whatsoever, whether or not $E_{t-1}[u_t u_t']$ is diagonal
  - So, we could treat e.g. nonlinear VAR-like models
- What if $E_{t-1}\left[u_t^2\right]$ (or $E_{t-1}[u_t u_t']$) depends on $y_{t-1}, \ldots, y_{t-p+1}$?

Time-series models

- So far, we have considered $y_t = f(x_t) + u_t$
- What if x_t includes $y_{t-1}, \ldots, y_{t-p+1}$?
  - Recall Bayesian interpretation and write $p(y) = p(y_1, \ldots, y_p) \cdot p(y_{p+1}|y_p, \ldots, y_1) \cdots p(y_T|y_{T-1}, \ldots, y_1)$
  - Nothing changes, provided that we condition on p initial values
  - Even stationarity does not seem to be an issue
- What if y_t is multivariate?
  - No problem whatsoever, whether or not $E_{t-1}[u_t u_t']$ is diagonal
  - So, we could treat e.g. nonlinear VAR-like models
- What if $E_{t-1}\left[u_t^2\right]$ (or $E_{t-1}[u_t u_t']$) depends on $y_{t-1}, \ldots, y_{t-p+1}$?
  - Does not seem analytically tractable
  - Work in progress, using an iterative approach to estimate mean and log-volatility equations

Factor models

Factor models

- In the paper: simulation study for linear and nonlinear factor models

Factor models

- In the paper: simulation study for linear and nonlinear factor models
- We compare kernel ridge regression to
  - PC: regression of y on the principal components (PCs) of X
  - PC^2: regression of y on the PCs of X and the squares of these PCs (Bai and Ng, 2008)
  - SPC: regression of y on the PCs of $\begin{pmatrix} X & X^2 \end{pmatrix}$ (Bai and Ng, 2008)

Factor models

- In the paper: simulation study for linear and nonlinear factor models
- We compare kernel ridge regression to
  - PC: regression of y on the principal components (PCs) of X
  - PC^2: regression of y on the PCs of X and the squares of these PCs (Bai and Ng, 2008)
  - SPC: regression of y on the PCs of $\begin{pmatrix} X & X^2 \end{pmatrix}$ (Bai and Ng, 2008)
- Main findings:
  - Kernels perform competitively for "standard" DGPs, and better for nonstandard DGPs
  - Gaussian kernel is a "catch-all" method: never performs poorly; performs very well for "difficult" DGPs

Other cross-sectional models

Other cross-sectional models

- In the companion paper: simulation study for wide range of models, to study the effects of choosing "wrong" kernel or tuning parameters

Other cross-sectional models

- In the companion paper: simulation study for wide range of models, to study the effects of choosing "wrong" kernel or tuning parameters
- Main findings:
  - Rules of thumb for selecting tuning parameters work well
  - Gaussian kernel acts as a "catch-all" method again, moreso than polynomial kernels

Data

Data

- 132 U.S. macroeconomic variables, 1959:1-2010:1, monthly observations, transformed to stationarity (Stock and Watson, 2002)

Data

- 132 U.S. macroeconomic variables, 1959:1-2010:1, monthly observations, transformed to stationarity (Stock and Watson, 2002)

- We forecast four key series: Industrial Production, Personal Income, Manufacturing & Trade Sales, and Employment

Data

- 132 U.S. macroeconomic variables, 1959:1-2010:1, monthly observations, transformed to stationarity (Stock and Watson, 2002)

- We forecast four key series: Industrial Production, Personal Income, Manufacturing & Trade Sales, and Employment

- h-month-ahead out-of-sample forecasts of annualized h-month growth rate $y_{t+h}^{h} = \frac{1200}{h} \ln\left(y_{t+h}/y_t\right)$, for $h = 1, 3, 6, 12$

Data

- 132 U.S. macroeconomic variables, 1959:1-2010:1, monthly observations, transformed to stationarity (Stock and Watson, 2002)
- We forecast four key series: Industrial Production, Personal Income, Manufacturing & Trade Sales, and Employment
- h-month-ahead out-of-sample forecasts of annualized h-month growth rate $y_{t+h}^{h} = \frac{1200}{h} \ln\left(y_{t+h}/y_t\right)$, for $h = 1, 3, 6, 12$
- Rolling estimation window of length 120 months

Competing models

Competing models

- Standard benchmarks: mean, random walk, AR

Competing models

- Standard benchmarks: mean, random walk, AR
- DI-AR-Lag framework (Stock and Watson, 2002): regressors are lagged y_t and lagged factors

Competing models

- Standard benchmarks: mean, random walk, AR
- DI-AR-Lag framework (Stock and Watson, 2002): regressors are lagged y_t and lagged factors
  - Factors extracted using PC, PC^2, or SPC
  - Lag lengths and number of factors reselected for each forecast by minimizing BIC

Competing models

- Standard benchmarks: mean, random walk, AR
- DI-AR-Lag framework (Stock and Watson, 2002): regressors are lagged y_t and lagged factors
  - Factors extracted using PC, PC^2, or SPC
  - Lag lengths and number of factors reselected for each forecast by minimizing BIC
- Kernel ridge regression: same setup, but with lagged factors replaced by φ(lagged x_t)

Competing models

- Standard benchmarks: mean, random walk, AR
- DI-AR-Lag framework (Stock and Watson, 2002): regressors are lagged y_t and lagged factors
  - Factors extracted using PC, PC^2, or SPC
  - Lag lengths and number of factors reselected for each forecast by minimizing BIC
- Kernel ridge regression: same setup, but with lagged factors replaced by φ(lagged x_t)
  - Polynomial kernels of degree 1 and 2, and the Gaussian kernel
  - Lag lengths, λ and σ selected by leave-one-out cross-validation

MSPEs for Industrial Production and Personal Income

Forecast method	Industrial Production $h=1$	$h=3$	$h=6$	$h=12$	Personal Income $h=1$	$h=3$	$h=6$	$h=12$
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

MSPEs for Industrial Production and Personal Income

- Simple PC performs better than its nonlinear extensions
- Kernel methods perform even slightly better
- "Infinite-dimensional", smooth Gaussian kernel is a safe choice
- Good results at all horizons

MSPEs for Manufacturing & Trade Sales and Employment

Forecast method	Manufacturing & Trade Sales $h=1$	$h=3$	$h=6$	$h=12$	Employment $h=1$	$h=3$	$h=6$	$h=12$
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

MSPEs for Manufacturing & Trade Sales and Employment

- Small losses at all horizons
- Linear model is apparently sufficient here, but Gaussian KRR continues to yield adequate results
- Both PC and KRR work very well
- PC outperforms all other methods

A closer look at performance

A closer look at performance

- So, KRR performs worse than PC only if PC performs very well

A closer look at performance

- So, KRR performs worse than PC only if PC performs very well
- To see if this result also holds over time, we computed mean squared prediction errors for each ten-year window separately

A closer look at performance

- So, KRR performs worse than PC only if PC performs very well
- To see if this result also holds over time, we computed mean squared prediction errors for each ten-year window separately
- All methods yield larger errors in more volatile periods

A closer look at performance

- So, KRR performs worse than PC only if PC performs very well
- To see if this result also holds over time, we computed mean squared prediction errors for each ten-year window separately
- All methods yield larger errors in more volatile periods
- However: smaller *relative* errors in more volatile periods

A closer look at performance

- So, KRR performs worse than PC only if PC performs very well
- To see if this result also holds over time, we computed mean squared prediction errors for each ten-year window separately
- All methods yield larger errors in more volatile periods
- However: smaller *relative* errors in more volatile periods
- KRR produces more volatile relative errors than PC
  $\Rightarrow$ KRR most valuable in turmoil periods, including 2008-9 crisis

Forecast encompassing regressions

Forecast encompassing regressions

- Forecast encompassing regression:

$$y^h_{t+h} = \alpha\, \hat{y}^{h,\text{ PC or KRR}}_{t+h|t} + (1-\alpha)\, \hat{y}^{h,\text{ AR}}_{t+h|t} + u^h_{t+h}$$

Forecast encompassing regressions

- Forecast encompassing regression:

$$y^h_{t+h} = \alpha \, \hat{y}^{h,\text{ PC or KRR}}_{t+h|t} + (1-\alpha) \, \hat{y}^{h,\text{ AR}}_{t+h|t} + u^h_{t+h}$$

- Hypotheses of interest: $\alpha = 0$ and $\alpha = 1$

Forecast encompassing regressions

- Forecast encompassing regression:

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

- Hypotheses of interest: $\alpha = 0$ and $\alpha = 1$

- Across all series and horizons, $\alpha = 0$ is strongly rejected for PC and for all KRR forecasts

Forecast encompassing regressions

- Forecast encompassing regression:

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

- Hypotheses of interest: $\alpha = 0$ and $\alpha = 1$
- Across all series and horizons, $\alpha = 0$ is strongly rejected for PC and for all KRR forecasts
- In many cases, $\alpha = 1$ cannot be rejected

Forecast encompassing regressions

- Forecast encompassing regression:

$$y^h_{t+h} \;=\; \alpha\, \hat{y}^{h,\text{ PC or KRR}}_{t+h|t} \;+\; (1-\alpha)\; \hat{y}^{h,\text{ AR}}_{t+h|t} \;+\; u^h_{t+h}$$

- Hypotheses of interest: $\alpha = 0$ and $\alpha = 1$
- Across all series and horizons, $\alpha = 0$ is strongly rejected for PC and for all KRR forecasts
- In many cases, $\alpha = 1$ cannot be rejected
- Thus, PC and KRR forecasts encompass AR forecasts

Forecast encompassing regressions

Forecast encompassing regressions

- Also compare kernels and PC:

$$y^h_{t+h} \;=\; \alpha\, \hat{y}^{h,\ \mathrm{KRR}}_{t+h|t} \;+\; (1-\alpha)\, \hat{y}^{h,\ \mathrm{PC}}_{t+h|t} \;+\; u^h_{t+h}$$

Forecast encompassing regressions

- Also compare kernels and PC:

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

- In most cases, we reject both $\alpha = 0$ and $\alpha = 1$

Forecast encompassing regressions

- Also compare kernels and PC:

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

- In most cases, we reject both $\alpha = 0$ and $\alpha = 1$
- That is, $0 < \alpha < 1$: KRR and PC forecasts are complements

Conclusions

Conclusions

- Kernel ridge regression provides a natural way of dealing with high-dimensionality and nonlinearity

Conclusions

- Kernel ridge regression provides a natural way of dealing with high-dimensionality and nonlinearity
- It can also handle time-series models with constant conditional volatilities and correlations, even if they are nonstationary

Conclusions

- Kernel ridge regression provides a natural way of dealing with high-dimensionality and nonlinearity
- It can also handle time-series models with constant conditional volatilities and correlations, even if they are nonstationary
- Selection of kernel and tuning parameters can be fully automated: easy-to-use black-box implementation for nonlinear forecasting

Conclusions

- Kernel ridge regression provides a natural way of dealing with high-dimensionality and nonlinearity
- It can also handle time-series models with constant conditional volatilities and correlations, even if they are nonstationary
- Selection of kernel and tuning parameters can be fully automated: easy-to-use black-box implementation for nonlinear forecasting

- Macro forecasting: KRR outperforms more traditional methods

Conclusions

- Kernel ridge regression provides a natural way of dealing with high-dimensionality and nonlinearity
- It can also handle time-series models with constant conditional volatilities and correlations, even if they are nonstationary
- Selection of kernel and tuning parameters can be fully automated: easy-to-use black-box implementation for nonlinear forecasting

- Macro forecasting: KRR outperforms more traditional methods
- Best forecast performance in turmoil periods

Conclusions

- Kernel ridge regression provides a natural way of dealing with high-dimensionality and nonlinearity
- It can also handle time-series models with constant conditional volatilities and correlations, even if they are nonstationary
- Selection of kernel and tuning parameters can be fully automated: easy-to-use black-box implementation for nonlinear forecasting

- Macro forecasting: KRR outperforms more traditional methods
- Best forecast performance in turmoil periods
- The "smooth" Gaussian kernel generally performs best

Current research

Current research

- Examine a wider range of kernel functions

Current research

- Examine a wider range of kernel functions
  - So far, Gaussian kernel holds up very well

Current research

- Examine a wider range of kernel functions
  - So far, Gaussian kernel holds up very well
- Extend the methodology to models with time-varying volatility

Current research

- Examine a wider range of kernel functions
  - So far, Gaussian kernel holds up very well
- Extend the methodology to models with time-varying volatility
  - This will enable applications to financial data