Contents

1 Forecasting discount curves with Kernel Regularized Least Squares

1.1 Introduction

In this chapter, we apply Kernel Regularized Least Squares (KRLS) learning methods to Yield Curve forecasting. By 'Yield Curve', we actually mean a discount curves. That is, we consider that the curves used in theexamples do not include counterparty credit risk, and we focus on the forecasting problem. Two types of formulations of the forecasting problem are tested. One relying on the popular Dynamic Nelson-Siegel (DNS) framework from [DL06], and another one, in which we apply the KRLS directly to the Yield Curve observation dates and time to maturities, to model the spot rates.

In the DNS framework [DL06], each cross-section of yields observed over time is fitted by using the Nelson-Siegel [NS87] model. This cross-section fitting produces three time series of parameters (more details in the next section) representing the evolution of the level, slope, and curvature of the Yield Curve. The KRLS model is applied to forecasting the time series of parameters, using a technique which is similar to the one described in [Ext+16]. And to finish, the forecast obtained for the trivariate time series are plugged into the Nelson-Siegel model, to deduce forecast for the cross-sections of yields.

The second approach is a machine learning/data-driven one, in which we put no specific constraint on the model to reproduce the specific Yield Curve stylized facts. The regularization parameters inherent to the KRLS model will act as implicit constraints, that cause the model to converge as close as possible to reproducing these stylized facts. In this latter approach, we are mostly interested in the model with the *best* out-of-sample error. As a consequence, the technique as is, is probably less adapted than the former framework based on DNS (in its arbitrage-free version) to no-arbitrage pricing (if no-arbitrage pricing is required).

To introduce KRLS, we start by describing the ridge regression [HK70] and the *kernel trick* applied to ridge regression. Then, we make a link between this kernel ridge regression and KRLS. In a ridge regression setting, we want to explain an observed variable $y \in \mathbb{R}^n$, as a linear function of p predictors stored in a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. For $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, p\}$ we have:

$$X_{ij} =: \mathbf{x}_i^{(j)}$$

We will denote the i^{th} row of $\mathbf{X}$ as $\mathbf{x}_i$, and its j^{th} column as $\mathbf{x}^{(j)}$. Hence, we are searching for the parameters $\beta = (\beta_1, \dots, \beta_p)^T$ verifying:

$$ArgMin_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \beta \right)^2$$

under the constraint

$$||\beta||_2^2 \leq s$$

The solution to this problem is given directly by the formula:

$$\hat{\beta} = \left(\mathbf{X}^T \mathbf{X} + \lambda I_{p \times p} \right)^{-1} \mathbf{X}^T y$$

where λ is a Lagrange multiplier having a unique correspondance with s, and a regularization parameter preventing the model from overfitting the observed data contained in y. In the case where we want to explain y as a function Φ of the predictors, we have a similar expression:

$$\hat{\beta} = \left(\Phi(\mathbf{X})^T \Phi(\mathbf{X}) + \lambda I_{p \times p} \right)^{-1} \Phi(\mathbf{X})^T y$$

where:

$$\Phi(\mathbf{X})_{ij} = \Phi(\mathbf{x}_i^{(j)})$$

Now, by using the Woodbury identity (cite Gene H. Golub and Charles F. van Loan. Matrix Computations and cite Max Welling The Kalman filter, Lecture Note) for $\mathbf{P}$ and $\mathbf{R}$ positive definite

$$\left(\mathbf{P}^{-1} + \mathbf{B}^T \mathbf{R}^{-1} \mathbf{B} \right)^{-1} \mathbf{B}^T \mathbf{R}^{-1} = \mathbf{P} \mathbf{B}^T \left(\mathbf{B} \mathbf{P} \mathbf{B}^T + \mathbf{R} \right)^{-1}$$

The solution to the ridge regression problem can be re-written as:

$$\hat{\beta} = \Phi(\mathbf{X})^T \left(\Phi(\mathbf{X}) \Phi(\mathbf{X})^T + \lambda I_{n \times n} \right)^{-1} y$$

This relationship can be useful in the case where $n << p$. That is, when there is a high number of predictors compared to the number of observations (cite Exterkate (2016)). Indeed, with this new relationship, we are no longer inverting a $p \times p$ matrix, but a $n \times n$ matrix. That's the *kernel trick*. And if some new observations arrive, and are stored in $\mathbf{X}^*$, the new values predicted by the model will be given by:

$$y^* = \Phi(\mathbf{X}^*)\hat{\beta} = \Phi(\mathbf{X}^*)\Phi(\mathbf{X})^T \left(\Phi(\mathbf{X})\Phi(\mathbf{X})^T + \lambda I_{n \times n}\right)^{-1} y$$

Which we re-write as:

$$y^* = \mathbf{K}^* \left(\mathbf{K} + \lambda I_{n \times n}\right)^{-1} y$$

$\mathbf{K}$ is a *kernel*; the empirical covariance matrix of $\Phi(\mathbf{X})^T$ (modulo a $1/p$ factor), in the case where the rows of $\Phi(\mathbf{X})$ are centered. Now, in the case of KRLS, the problem we are trying to solve is:

$$ArgMin_{c \in \mathbb{R}^n} \sum_{i=1}^{n} \left(y_i - K_i^T c\right)^2$$

where K_i is the i^{th} row of $\mathbf{K}$, with:

$$\mathbf{K}_{ij} =: K(\mathbf{x}_i, \mathbf{x}_j) = f(||\mathbf{x}_i - \mathbf{x}_j||_1) \;\; or \;\; f(||\mathbf{x}_i - \mathbf{x}_j||_2)$$

As described in (cite Halmueller & Hazlett (2013)), two approaches can be used to interpret/motivate the KRLS: a similarity-based view and the superposition of Gaussians view. Here, we refer only to the first one, which is the most intuitive to us, and is also the one described in Ferwerda, Hainmueller et Hazlett (2017). Indeed, here, the i^{th} observation y_i is explained as a linear combination of functions measuring the similarity/dissimilarity between its characteristics gathered in $\mathbf{x}_i$, and the other observations from the training set, $\mathbf{x}_j$, $j \neq i$:

$$y_i = \sum_{j=1}^{n} c_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{1.1}$$

But again, to prevent the model from overfitting the observed data and not being able to generalize, we need to constrain the parameters c under a certain norm defined by the matrix $\mathbf{K}$ (cite Hofmann, Schoelkopf, Smola (2008)):

$$||c||_K^2 = c^T \mathbf{K} c \leq s$$

The solution to this new problem is:

$$\hat{c} = (\mathbf{K} + \lambda I_{n \times n})^{-1} \mathbf{y}$$

where λ is a Lagrange multiplier having a unique correspondance with s, and a regularization parameter. And for new observations arriving for the model, we have a solution which is identical to the one that we had for kernel ridge regression:

$$y^* = \mathbf{K}^* (\mathbf{K} + \lambda I_{n \times n})^{-1} y$$

Many other types of kernels could be envisaged for $\mathbf{K}$, allowing to take into account nonlinearities and the various complexities of the covariance structure. One of the most popular kernels is the Gaussian kernel, also called *squared exponential* kernel, defined for $i < j$ by:

$$K_{Gauss}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||_2^2}{2l^2}\right)$$

where l is a characteristic length-scale controlling the distance between peaks of the covariance function. σ^2 is the marginal variance, obtained when $\mathbf{x}_i = \mathbf{x}_j$. Both l, σ^2 are used as the machine learning model's hyperparameters, along with the regularization parameter λ.

This kernel is however often judged as being too smooth for most typical optimization problems (cite Rasmussen et al.). Some other kernels that could be interesting for machine learning (cite Rasmussen et al.) belong to the Matérn class of covariance functions. If we define $r := ||\mathbf{x}_i - \mathbf{x}_j||_2$, the most used for machine learning problems (cite Rasmussen et al.) are:

$$\mathbf{K}_{ij} = K_{3/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{3}r}{l}\right) exp\left(-\frac{\sqrt{3}r}{l}\right)$$

and

$$\mathbf{K}_{ij} = K_{5/2}(r) = \sigma^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) exp\left(-\frac{\sqrt{5}r}{l}\right)$$

Another interesting feature of KRLS learning, is the possibility to derive estimators for the marginal effects of the covariates on the response. For example, (and like in Ferwerda, Hainmueller et Hazlett (2017)), since we have the relationship 1.1, we can write for a fixed $j_0 \in \{1, \ldots, p\}$ and for any $k \in \{1, \ldots, n\}$, :

$$\frac{\partial y_i}{\partial \mathbf{x}_k^{(j_0)}} = \sum_{j=1}^{n} c_j \frac{dK(\mathbf{x}_i, \mathbf{x}_j)}{d\mathbf{x}_k^{(j_0)}} = c_k \frac{\partial K(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{x}_k^{(j_0)}} \tag{1.2}$$

That's an approximation of how much of an increase (at the first order) we obtain in y_i, for a slight change in $\mathbf{x}_k^{(j_0)}$. An average marginal effect of the j_0^{th} covariate on the i^{th} observation of the response y can thus be obtained as:

$$\frac{1}{n} \sum_{k=1}^{n} \frac{\partial y_i}{\partial x_k^{(j_0)}} = \frac{1}{n} \sum_{k=1}^{n} c_k \frac{\partial K(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{x}_k^{(j_0)}} \tag{1.3}$$

In section 1.2, we derive formulas for the sensitivities of the response variable y_i to a change in the covariates $\mathbf{x}^{(j)}$. We do this for Gaussian, Matérn 3/2 and Matérn 5/2 kernels. These sensitivities can then be plugged into formula 1.3, in order to obtain average marginal effects of the covariates on the response.

1.2 Sensitivity of the response to a change in the covariates

We let:

$$r^2 := ||\mathbf{x}_i - \mathbf{x}_k||_2^2 = (\mathbf{x}_i - \mathbf{x}_k)^T (\mathbf{x}_i - \mathbf{x}_k)$$

Hence:

$$\frac{\partial r^2}{\partial \mathbf{x}_k^{(j_0)}} = -2\left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right)$$

And:

$$\frac{\partial r}{\partial \mathbf{x}_k^{(j_0)}} = \frac{1}{2r} \frac{\partial r^2}{\partial \mathbf{x}_k^{(j_0)}} = -\frac{1}{r}\left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right)$$

As a consequence:

- For the **Gaussian kernel**

$$(\mathbf{x}_i, \mathbf{x}_k) \mapsto \sigma^2 exp\left(-\frac{r^2}{2l^2}\right)$$

  We have:

$$\begin{aligned}
\frac{\partial K_{Gauss}(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{x}_k^{(j_0)}} &= -\frac{\sigma^2}{2l^2} exp\left(-\frac{r^2}{2l^2}\right) \frac{\partial r^2}{\partial \mathbf{x}_k^{(j_0)}} \\
&= -\frac{\sigma^2}{2l^2} exp\left(-\frac{r^2}{2l^2}\right) \left[-2\left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right)\right] \\
&= \frac{\sigma^2}{l^2} exp\left(-\frac{r^2}{2l^2}\right) \left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right) \\
&= \frac{\left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right)}{l^2} K_{Gauss}(\mathbf{x}_i, \mathbf{x}_k)
\end{aligned}$$

- For the **Matérn 3/2 kernel**

$$(\mathbf{x}_i, \mathbf{x}_k) \mapsto \sigma^2 \left(1 + \frac{\sqrt{3}r}{l}\right) exp\left(-\frac{\sqrt{3}r}{l}\right)$$

  We have:

$$\begin{aligned}
\frac{\partial K_{3/2}(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{x}_k^{(j_0)}} &= \sigma^2 \frac{\sqrt{3}}{l} exp\left(-\frac{\sqrt{3}}{l}r\right) \left[\frac{\partial r}{\partial \mathbf{x}_k^{(j_0)}} - \frac{\partial r}{\partial \mathbf{x}_k^{(j_0)}}\left(1 + \frac{\sqrt{3}}{l}r\right)\right] \\
&= \sigma^2 \frac{\sqrt{3}}{l} \frac{\partial r}{\partial \mathbf{x}_k^{(j_0)}} exp\left(-\frac{\sqrt{3}}{l}r\right) \left[1 - \left(1 + \frac{\sqrt{3}}{l}r\right)\right] \\
&= \sigma^2 \frac{\sqrt{3}}{l} exp\left(-\frac{\sqrt{3}}{l}r\right) \frac{\sqrt{3}\left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right)}{l}
\end{aligned}$$

- For the **Matérn 5/2 kernel**

$$(\mathbf{x}_i, \mathbf{x}_k) \mapsto \sigma^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) exp\left(-\frac{\sqrt{5}r}{l}\right)$$

  We have:

$$
\begin{aligned}
\frac{\partial K_{5/2}(\mathbf{x}_i, \mathbf{x}_k)}{\partial \mathbf{x}_k^{(j_0)}} &= \sigma^2 exp\left(-\frac{\sqrt{5}}{l} r\right) \left[\left(\frac{\sqrt{5}}{l} \frac{\partial r}{\partial \mathbf{x}_k^{(j_0)}} + \frac{5}{3l^2} \frac{\partial r^2}{\partial \mathbf{x}_k^{(j_0)}}\right) - \frac{\sqrt{5}}{l} \frac{\partial r}{\partial \mathbf{x}_k^{(j_0)}} \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right)\right. \\
&= \sigma^2 exp\left(-\frac{\sqrt{5}}{l} r\right) \left[\frac{5}{3l^2} \frac{\partial r^2}{\partial \mathbf{x}_k^{(j_0)}} - \frac{\sqrt{5}}{l} \frac{\partial r}{\partial \mathbf{x}_k^{(j_0)}} \left(\frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right)\right] \\
&= \sigma^2 exp\left(-\frac{\sqrt{5}}{l} r\right) \left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right) \left[-2\frac{5}{3l^2} + \frac{1}{r}\frac{\sqrt{5}}{l} \left(\frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right)\right] \\
&= \sigma^2 exp\left(-\frac{\sqrt{5}}{l} r\right) \frac{5}{3l^2} \left(1 + \frac{\sqrt{5}r}{l}\right) \left(\mathbf{x}_i^{(j_0)} - \mathbf{x}_k^{(j_0)}\right)
\end{aligned}
$$

1.3 Description of DNS-KRLS model

In the DNS framework ([DL06]), the spot interest rates observed at time t, for time to maturity τ are modeled as:

$$
R_t(\tau) = \alpha_{1,t} + \alpha_{2,t} \left(\frac{1 - e^{-\tau/\eta}}{e^{-\tau/\eta}}\right) + \alpha_{3,t} \left(\frac{1 - e^{-\tau/\eta}}{e^{-\tau/\eta}} - e^{-\tau/\eta}\right) \qquad (1.4)
$$

If these spot interest rates $R_t(\tau)$ are observed at increasing dates $t = t_1 < \ldots < t_n$, for increasing time to maturities $\tau = \tau_1 < \ldots < \tau_p$, the factor loadings are the vectors of length p:

$$
(1, \ldots, 1)^T
$$

and

$$
\left(\frac{1 - e^{-\tau_1/\eta}}{e^{-\tau_1/\eta}}, \ldots, \frac{1 - e^{-\tau_p/\eta}}{e^{-\tau_p/\eta}}\right)^T
$$

and

$$
\left(\frac{1 - e^{-\tau_1/\eta}}{e^{-\tau_1/\eta}} - e^{-\tau_1/\eta}, \ldots, \frac{1 - e^{-\tau_p/\eta}}{e^{-\tau_p/\eta}} - e^{-\tau_p/\eta}\right)^T
$$

They are used to represent respectively the level, slope, and curvature of the Yield Curve. Estimations of $\alpha_{i,t}, i = 1, \ldots, 3$ are obtained for each cross-section of yields (that is, for each fixed date t) by taking a fixed η, and doing a least squares regression of the spot rates observed at time t on these factor loadings. The three time series $(\alpha_{i,t})_t, i = 1, \ldots, 3$ associated to the loadings for each cross-section of yields, are those that we wish to forecast simultaneously, by using KRLS learning.

For doing this, we store the three time series in a response matrix $\mathbf{Y}$, and their lags in a matrix of predictors $\mathbf{X}$. We will then have, for new observations stored in $\mathbf{X}^*$:

$$\mathbf{Y}^* = \mathbf{K}^* \left(\mathbf{K} + \lambda I_{n\times n}\right)^{-1} \mathbf{Y}$$

We consider $p \in \mathbb{N}^*$ time series $(X_t^{(j)})_{t\geq 0}, j = 1, \ldots, p$, observed at $n \in \mathbb{N}^*$ discrete dates. We are interested in obtaining simultaneous forecasts of the p time series at time $n+h$, $h \in \mathbb{N}^*$, by allowing each of the p variables to be influenced by the others (in the spirit of VAR models, see [Lüt05]).

For this purpose, we use $k < n$ lags of each of the observed p time series. The output variables to be explained are:

$$Y^{(j)} = \left(X_n^{(j)}, \ldots, X_{k+1}^{(j)}\right)^T \tag{1.5}$$

for $j \in \{1, \ldots, p\}$. Where $X_n^{(j)}$ is the most contemporaneously observed value of the j^{th} time series, and $X_{k+1}^{(j)}$ was observed k dates earlier in time for $(X_t^{(j)})_{t\geq 0}$. These output variables are stored in:

$$\mathbf{Y} \in \mathbb{R}^{(n-k)\times p}$$

and the predictors are stored in:

$$\mathbf{X} \in \mathbb{R}^{(n-k)\times (k\times p)}$$

where $\mathbf{X}$ consists in p blocks of k lags, for each one of the observed p time series. For example, the j_0^{th} block of $\mathbf{X}$, for $j_0 \in \{1, \ldots, p\}$ contains in columns:

$$\left(X_{n-i}^{(j_0)}, \ldots, X_{k+1-i}^{(j_0)}\right)^T \tag{1.6}$$

with $i \in \{1, \ldots, k\}$. If we consider $p = 2$ time series $(X_{t_1}^{(1)}, \ldots, X_{t_5}^{(1)})$ and $(X_{t_1}^{(2)}, \ldots, X_{t_5}^{(2)})$ observed at $n = 5$ dates $t_1 < \ldots < t_5$, with $k = 2$ lags, the response variables are stored in:

$$\mathbf{Y} = \begin{pmatrix} X_{t_5}^{(1)} & X_{t_5}^{(2)} \\ X_{t_4}^{(1)} & X_{t_4}^{(2)} \\ X_{t_3}^{(1)} & X_{t_3}^{(2)} \end{pmatrix}$$

The predictors are stored in:

$$\mathbf{X} = \begin{pmatrix} X_{t_4}^{(1)} & X_{t_3}^{(1)} & X_{t_4}^{(2)} & X_{t_3}^{(2)} \\ X_{t_3}^{(1)} & X_{t_2}^{(1)} & X_{t_3}^{(2)} & X_{t_2}^{(2)} \\ X_{t_2}^{(1)} & X_{t_1}^{(1)} & X_{t_2}^{(2)} & X_{t_1}^{(2)} \end{pmatrix}$$

It is also possible to add other regressors, such as dummy variables, or indicators of special events. In this situation, as discussed in [Ext+16], we can avoid the constraining of these dummy variables, in a Kernel ridge regression with unpenalized terms. Here, we consider only the inclusion of the observed time series' lags in the model.

1.4 Description of the KRLS model applied to observed dates and time to maturities

In this other framework, we consider that the response variable is the spot interest rate observed at time t, for time to maturity τ, $R(t, \tau)$. The predictors are the observation date, and the time to maturity. In this setting, we use the following weighted distance between the vectors (t, τ) in the Gaussian, Matérn 3/2 and Matérn 5/2 kernels:

$$r = \sqrt{\frac{||t_i - t_j||_2^2}{l_1^2} + \frac{||\tau_i - \tau_j||_2^2}{l_2^2}}$$

So that here, the spot rates values are explained as linear combination of distances between vectors of time to maturities and observation dates (t_i, τ_i) and (t_j, τ_j). In this setting, if we consider 10 spot rates observed at 2 dates $t_1 < t_2$ and 5 time to maturities $\tau_1, \dots, \tau_5$, the response variable is:

$$\mathbf{Y} = (R(t_1, \tau_1), \dots, R(t_1, \tau_5), R(t_2, \tau_1), \dots, R(t_2, \tau_5))^T$$

and the predictors are

$$\mathbf{X} = \begin{pmatrix} \tau_1 & t_1 \\ \vdots & \vdots \\ \tau_5 & t_1 \\ \tau_1 & t_2 \\ \vdots & \vdots \\ \tau_5 & t_2 \end{pmatrix}$$

If some new observations arrive at time t_3 in the model, these new observations will be stored in:

$$\mathbf{X}^* = \begin{pmatrix} \tau_1 & t_3 \\ \vdots & \vdots \\ \tau_5 & t_3 \end{pmatrix}$$

In this other setting, it is also possible to add other regressors such as dummy variables, or indicators of special events. Again, and as suggested in the previous section and in [Ext+16], we can avoid the constraining of these dummy variables, in a Kernel ridge regression with unpenalized terms. For example, if we wanted to add another indicator $(I_t)_t$ observed at times $t_1 < t_2$, we would have to consider the following matrix of predictors:

$$\mathbf{X} = \begin{pmatrix} \tau_1 & t_1 & I_{t_1} \\ \vdots & \vdots & \vdots \\ \tau_5 & t_1 & I_{t_1} \\ \tau_1 & t_2 & I_{t_2} \\ \vdots & \vdots & \vdots \\ \tau_5 & t_2 & I_{t_2} \end{pmatrix}$$

And another weighted distance in the Gaussian, Matérn 3/2 and Matérn 5/2 kernels, taking into account the new indicator $(I_t)_t$:

$$r = \sqrt{\frac{||t_i - t_j||_2^2}{l_1^2} + \frac{||\tau_i - \tau_j||_2^2}{l_2^2} + \frac{||I_{t_i} - I_{t_j}||_2^2}{l_3^2}}$$

In the next section, 1.5, we present the results obtained by the DNS-KRLS model from section 1.3 and the KRLS model from section 1.4 on a training/testing dataset[1].

1.5 Numerical examples

In this section, we present the results obtained by the DNS-KRLS model from section 1.3 and the KRLS model from section 1.4. The examples are not exhaustive benchmarks, but aim at illustrating the forecasting capabilities of the models.

We use calibrated discount rates data from Deutsche Bundesbank website, observed on a monthly basis, from the beginning of 2002 to the end 2015. There are 167

[1] for a further treatment, a validation set would be added, in order to verify that the models are not 'overtrained' on this training/testing set

Tab. 1.1: Summary of observed discount rates from Deutsche Bundesbank website, from 2002 to the end 2015

Maturity	Min	1st Qrt	Median	3rd Qrt	Max
1	-0.116	0.858	2.045	3.072	5.356
5	0.170	1.327	2.863	3.807	5.146
15	0.711	2.616	3.954	4.702	5.758
30	0.805	2.594	3.962	4.814	5.784

curves, observed at 50 maturities in the dataset. Only 15 time to maturities are used in these examples (in years): $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 15, 20, 25, 30$.

In figure 1.1, we present the data that we use, and table 1.1 contains a summary of these data; the minimum, maximum, median, first and third quartiles of the discount rates observed at given maturities. There are alternate cycles of increases and decreases of the discount rates, with generally a decreasing trend. Some of the discount rates, at the most recent dates, and lower maturities, are negative.

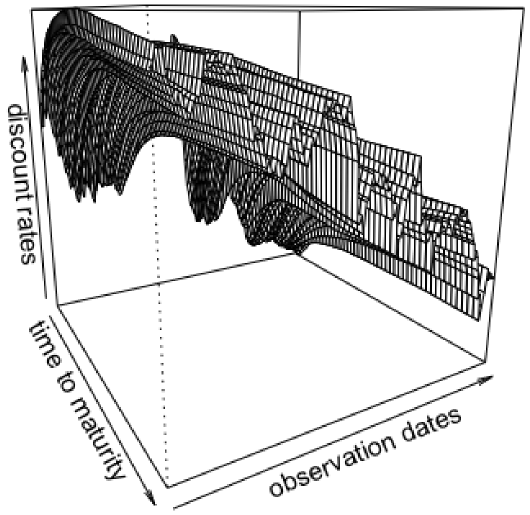

Fig. 1.1: Observed discount rates from Deutsche Bundesbank website, from 2002 to the end 2015

A rolling forecasting methodology (see [Ber+15]) is implemented in order to obtain the benchmarks between the models. A fixed 12 months/36 months-length window for training the model, and the following 12 months/36 months for testing, the origin of the training set is then advanced of 1 month, and the training/testing procedure is repeated. The measure of forecasting performance is the Root Mean Squared Error ($RMSE$).

We use similar grids for all the models, in order to ease the comparability of the results, and avoid too much manual tweaking of the hyperparameters and overtraining the available data.

Hence for both models, DNS-KRLS (from section 1.3) and KRLS (from section 1.4), we consider 5 values of σ (variance parameter), l, l_1, l_2 (length-scale parameters for Gaussian, Matérn 3/2 and Matérn 5/2 kernels) and λ (the regularization parameter for all the kernels) regularly spaced between $[10^{-2}, 10^{2}]$: $0.01, 0.1, 1, 10, 100$.

For the additional parameter η in the DNS-KRLS model, we use 5 values comprised (regularly spaced) between the minimum of the observed time to maturities and the maximum of the observed time to maturities (on $[1, 30]$): $1, 8.25, 15.5, 22.75, 30$.

1.5.1 Cross-validation results

The results obtained after the cross-validation procedure are reported in table 1.2 and 1.3.

Tab. 1.2: Average out-of-sample RMSE for training set length = 12 months and test set length = 12 months

Model	σ	l	l_1	l_2	λ	η	**RMSE**
Gaussian	10	-	0.01	10	10	-	0.5839150
Matérn 3/2	100	-	1	100	0.1	-	**0.5136373**
Matérn 5/2	1	-	0.01	10	0.1	-	0.5781184
DNS-Gaussian	0.1	10	-	-	0.01	15.5	0.6041652
DNS-Matérn 3/2	100	1	-	-	0.01	22.75	**0.6038667**
DNS-Matérn 5/2	100	1	-	-	0.01	15.5	0.6041580

We observe that no matter the length of the training/testing window (either 12 months or 36 months), or the method employed (either DNS-KRLS or KRLS), the Matérn 3/2 kernel performs better than the other models. It is even performing better on this specific problem with a KRLS model, considering the similarities between vectors of time to maturities and observation dates. In the next section 1.5.2, we examine these results further, by looking at the projected discount/discount factors'/discrete forward curves.

Tab. 1.3: Average out-of-sample RMSE for training set length = 36 months and test set length = 36 months

Model	σ	l	l_1	l_2	λ	η	**RMSE**
Gaussian	10	-	0.1	10	100	-	1.170690
Matérn 3/2	100	-	1	10	0.01	-	**1.003246**
Matérn 5/2	10	-	10	100	10	-	1.134833
DNS-Gaussian	0.01	0.01	-	-	0.01	30	1.264533
DNS-Matérn 3/2	0.01	0.01	-	-	0.01	30	**1.264533**
DNS-Matérn 5/2	0.01	0.01	-	-	0.01	30	1.264533

1.5.2 Implied forecast term-structure of discrete forward rates

1.5.3 Sensitivities

1.6 Conclusion

Bibliography

[Ber+15] Christoph Bergmeir, Rob J Hyndman, Bonsoo Koo, et al. „A Note on the Validity of Cross-Validation for Evaluating Time Series Prediction". In: *Monash University, Department of Econometrics and Business Statistics, Tech. Rep.* (2015) (cit. on p. 11).

[Bon+15] F Bonnin, F Combes, F Planchet, and M Tammar. „Un modèle de projection pour des contrats de retraite dans le cadre de l'ORSA". In: *Bulletin Français d'Actuariat* 14 (2015), pp. 107–129.

[BT97] Phelim P Boyle and Ken Seng Tan. „Quasi-Monte Carlo methods". In: *International AFIR Colloquium Proceedings, Australia*. Vol. 1. 1997, pp. 1–24.

[Car98] Rich Caruana. „Multitask learning". In: *Learning to learn*. Springer, 1998, pp. 95–133.

[Cha+92] Kanad Chakraborty, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. „Forecasting the behavior of multivariate time series using neural networks". In: *Neural networks* 5.6 (1992), pp. 961–970.

[Cor09] Thomas H Cormen. *Introduction to algorithms*. MIT press, 2009.

[DC10] Satchidananda Dehuri and Sung-Bae Cho. „A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN". In: *Neural Computing and Applications* 19.2 (2010), pp. 187–205.

[DL06] Francis X Diebold and Canlin Li. „Forecasting the term structure of government bond yields". In: *Journal of econometrics* 130.2 (2006), pp. 337–364 (cit. on pp. 1, 7).

[DR13] Francis X Diebold and Glenn D Rudebusch. *Yield Curve Modeling and Forecasting: The Dynamic Nelson-Siegel Approach*. Princeton University Press, 2013.

[DS15] Christophe Dutang and Petr Savicky. *randtoolbox: Generating and Testing Random Numbers*. R package version 1.17. 2015.

[Ext+16] Peter Exterkate, Patrick JF Groenen, Christiaan Heij, and Dick van Dijk. „Nonlinear forecasting with many predictors using kernel ridge regression". In: *International Journal of Forecasting* 32.3 (2016), pp. 736–753 (cit. on pp. 1, 9, 10).

[HK08] Robin Hyndman and Yeasmin Khandakar. „Automatic time series forecasting: The forecast Package for R". In: *Journal of Statistical Software* 27.3 (2008), pp. 1–22.

[HK70] Arthur E Hoerl and Robert W Kennard. „Ridge regression: Biased estimation for nonorthogonal problems“. In: *Technometrics* 12.1 (1970), pp. 55–67 (cit. on p. 1).

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. „Long short-term memory“. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[JK08] S Joe and F Kuo. *Notes on generating Sobol sequences*. `http://web.maths.unsw.edu.au/~fkuo/sobol/joe-kuo-notes.pdf`. 2008.

[Jür00] Manuela Jürgens. *LaTeX: eine Einführung und ein bisschen mehr*. FernUniversität Gesamthochschule in Hagen, 2000.

[Jür95] Manuela Jürgens. *LaTeX: Fortgeschrittene Anwendungen*. FernUniversität Gesamthochschule in Hagen, 1995.

[KJUM11] Markus Kohm and Jens-Uwe-Morawski. *KOMA-Script: Die Anleitung*. 2011.

[Lüt05] Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005 (cit. on p. 8).

[Mie11a] André Miede. *A Classic Thesis Style: An Homage to The Elements of Typographic Style*. 2011.

[Nie92] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. SIAM, 1992.

[NS87] Charles R Nelson and Andrew F Siegel. „Parsimonious modeling of yield curves“. In: *Journal of business* (1987), pp. 473–489 (cit. on p. 1).

[Pan12] Alan Pankratz. *Forecasting with dynamic regression models*. Vol. 935. John Wiley & Sons, 2012.

[Pao+94] Yoh-Han Pao, Gwang-Hoon Park, and Dejan J Sobajic. „Learning and generalization characteristics of the random vector functional-link net“. In: *Neurocomputing* 6.2 (1994), pp. 163–180.

[Pen55] Roger Penrose. „A generalized inverse for matrices“. In: *Mathematical proceedings of the Cambridge philosophical society*. Vol. 51. 03. Cambridge Univ Press. 1955, pp. 406–413.

[Pfa+08] Bernhard Pfaff et al. „VAR, SVAR and SVEC models: Implementation within R package vars“. In: *Journal of Statistical Software* 27.4 (2008), pp. 1–32.

[Ren+16] Ye Ren, PN Suganthan, N Srikanth, and Gehan Amaratunga. „Random vector functional link network for short-term electricity load demand forecasting“. In: *Information Sciences* 367 (2016), pp. 1078–1093.

[Rum+88] DE Rumelhart, GE Hinton, and RJ Williams. „Learning internal representations by error propagation“. In: *Neurocomputing: foundations of research*. MIT Press. 1988, pp. 673–695.

[Sch+92] Wouter F Schmidt, Martin A Kraaijveld, and Robert PW Duin. „Feedforward neural networks with random weights“. In: *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*. IEEE. 1992, pp. 1–4.

[Wic16] Hadley Wickham. *ggplot2: elegant graphics for data analysis*. Springer, 2016.

[ZS16] Le Zhang and PN Suganthan. „A comprehensive evaluation of random vector functional link networks“. In: *Information Sciences* 367 (2016), pp. 1094–1105.

[App10a] Apple Inc. *Keynote ’09 User Guide*. Apple Inc., 2010.

[App10b] Apple Inc. *Numbers ’09 User Guide*. Apple Inc., 2010.

[App10c] Apple Inc. *Pages ’09 User Guide*. Apple Inc., 2010.

Webseiten

[@Mie11b] André Miede. *A Classic Thesis Style by André Miede*. 2011. URL: `http://www.miede.de/index.php?page=classicthesis` (visited on May 27, 2011).

[@Fre10] Free Software Foundation, Inc. *GNU General Public License*. 2010. URL: `http://www.gnu.org/licenses/gpl.html` (visited on May 27, 2011).

List of Figures

List of Tables

Colophon

This thesis was typeset with LaTeX 2$_{\varepsilon}$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.

Declaration

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

City, August 26, 2015

Ricardo Langner