We illustrate the prediction of a real valued response when one of the covariates is a function using a widely analysed data set for quality control in the food industry. The data[1]contain samples of spectrometric curve of absorbances of 215 pieces of finely chopped meat, along with their water, fat and protein content. These data are recorded on a Tecator Infratec Food and Feed Analyzer working in the wavelength range 850 - 1050 nm by the Near Infrared Transmission (NIT) principle. Absorption data has not been measured continuously, but instead 100 distinct wavelengths were obtained. Figure 1 shows a sample of 10 such spectrometric curves.

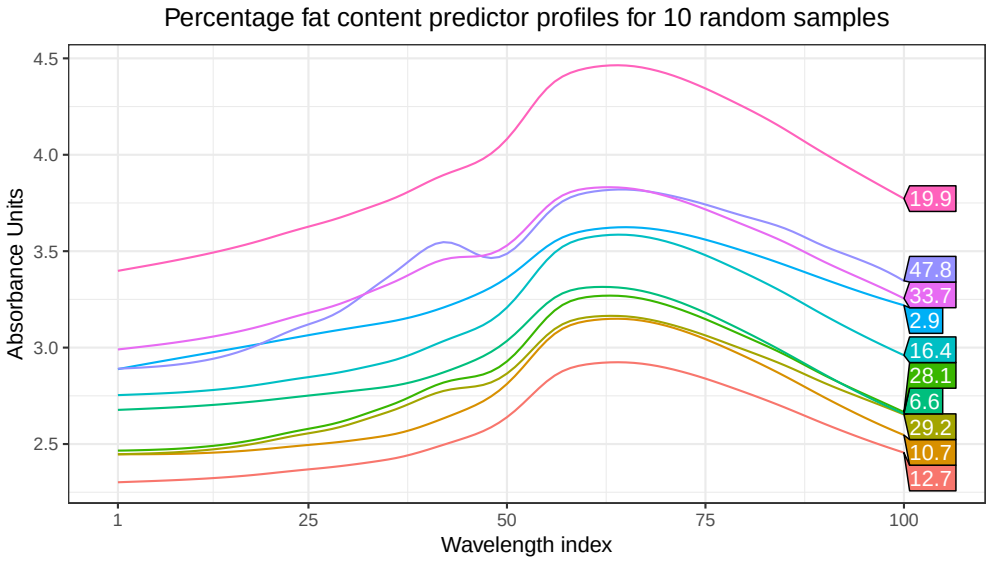

Figure 1: Sample of spectrometric curves used to predict fat content of meat. For each meat sample the data consists of a 100 channel spectrum of absorbances and the contents of moisture, fat (numbers shown in boxes) and protein measured in percent. The absorbance is $-\log 10$ of the transmittance measured by the spectrometer. The three contents, measured in percent, are determined by analytic chemistry.

For our analyses and many others' in the literature, the first 160 observations in the data set are used as a training sample for model fitting, and the remaining 55 observations as a test sample to evaluate the predictive performance of the fitted model. A summary of the various statistical methods applied to this data set, including various I-prior models, can be found in citebergsma2016. The focus here is to use the **iprior** package to fit various I-prior models to the Tecator data set.

Before we began, we preprocessed the spectral curves by taking their first differences . This leaves us with the 99-dimensional covariate, which is saved in the matrix object named `absorpTrain`. Our first modelling attempt is to estimate a linear effect

1. Necessary for functional covariates - approximation of the Sobolev-Hilbert space inner product

[1]Used with permission from Tecator (see `http://lib.stat.cmu.edu/datasets/tecator` for details). We used the version made available in the dataframe `tecator` from the R package **caret** for our analyses.

by regressing the responses `fatTrain` against only a single high-dimensional covariate `absorpTrain` using the canonical RKHS. The model is loaded as an `ipriorKernel` object as follows:

```
R> # Model 1: canonical RKHS (linear)
R> (mod1 <- kernL(y = fatTrain, absorpTrain))

##
## Sample size =  160
## Number of x variables, p =  1
## Number of scale parameters, l =  1
## Number of interactions =  0
##
## Info on H matrix:
##
## List of 1
##  $ absorpTrain: Canonical [1:160, 1:160] 0.000254 0.0003 -0.000231 -..
```

Here, we have used the non-formula syntax because each object after the `y` argument is treated as a single covariate, even if it is multi-dimensional, i.e., a matrix. We could have also used the formula syntax and used the `model` option `one.lam = TRUE`. Note that the canonical RKHS is used by default.

Our second and third model uses a polynomial-type construction of the canonical RKHS, which allows us to add quadratic and cubic terms of the spectral curves. The syntax is as before with the addition of `absorpTrain^b`, which element-wise raises the entries of the matrix `absorpTrain` to the power `b`. To date, the only method to fit these models parsimoniously in **iprior** is by using non-formula syntax with `model` option `order` to control the scale parameters of the RKHS. Both models only have a single parameter. Without specifying the `order` option, additional scale parameters would be fitted, one for each quadratic and cubic term.

```
R> # Model 2: canonical RKHS (quadratic)
R> mod2 <- kernL(y = fatTrain, absorpTrain, absorpTrain ^ 2,
+                model = list(order = c("1", "1^2")))

R> # Model 3: canonical RKHS (cubic)
R> mod3 <- kernL(y = fatTrain, absorpTrain, absorpTrain ^ 2, absorpTrain ^ 3,
+                model = list(order = c("1", "1^2", "1^3")))
```

Next, we fitted a smooth dependence of fat content on the spectrometric curves using the FBM RKHS. By default, the Hurst coefficient for the FBM RKHS is set to be 0.5. However, we can use the function `fbmOptim()` which is able to compute the maximum likelihood estimate for the Hurst coefficient.

```
R> # Model 4: FBM RKHS (default Hurst = 0.5)
R> mod4 <- kernL(y = fatTrain, absorpTrain, model = list(kernel = "FBM"))
```

Finally, we add an extra covariate (meat moisture content) which is assumed to have a linear effect on fat content. Doing so adds one extra parameter to the model. To specify multiple kernels, we need to include the `model` option `kernel = c("FBM", "Canonical")` to indicate the effect of the respective covariates on the response. This is verified by inspecting the `print` output of the `ipriorKernel` object, and indeed we see that there are now two scale parameters, and the kernel loader correctly assigns the FBM and canonical RKHS to the spectrometric curves and moisture content respectively.

```
R> # Model 5: FBM RKHS + extra covariate
R> (mod5 <- kernL(y = fatTrain, absorpTrain, waterTrain,
+                 model = list(kernel = c("FBM", "Canonical"))))

##
## Sample size =  160
## Number of x variables, p =  2
## Number of scale parameters, l =  2
## Number of interactions =  0
##
## Info on H matrix:
##
## List of 2
##  $ absorpTrain: FBM,0.5 [1:160, 1:160] 0.016192 -0.000775 -0.00346 -..
##  $ waterTrain : Canonical [1:160, 1:160] 10.9 58.9 -23.8 -29.7 18.2 ..
```

All of the above models were fitted using `ipriorOptim`, except for the last two model, where we used `fbmOptim` in order to obtain the maximum likelihood estimate for the Hurst coefficient of the FBM RKHS. Predicted values of the test data set can be obtained using the `predict` function

```
R> fatTestPredicted <- predict(mod1.fit, list(absorpTest))
R> head(fatTestPredicted)

## [1] 14.12268 15.85864 15.84706 21.59324 25.22315 26.57978
```

and the root mean squared error (RMSE) calculated for each of the models. It was noted that for some models, different EM starting values gave slightly different results, and we suspect this is a due to numerical issues with the computation of the variance of marginal I-prior distribution. Nonetheless, the predicted values, and hence the RMSE, remain fairly robust.

The results are summarised in Table 1. Models 1-3 have the same number of parameters, so a direct comparison can be done, with the model giving the highest likelihood value preferred. In this case, it is the model with a quadratic effect, giving a test RMSE of 1.23. Models with the FBM RKHS gave better prediction still. A smooth effect (Hurst = 0.5) yields a test RMSE of 0.67, and this is improved only slightly by using

the maximum likelihood estimate for the Hurst coefficient of 0.519. The best predictive model obtained was the final model, i.e., a smooth effect (Hurst = 0.934) with an additional covariate, giving a test RMSE of 0.68.

Model	I-prior effect	Log-likelihood	RMSE	
			Train	Test
1	Linear	−409.32	2.85	3.24
2	Quadratic	−279.64	0.72	1.23
3	Cubic	−301.26	0.99	1.65
4	Smooth (Hurst = 0.5)	−148.34	0.00	0.67
4a	Smooth (Hurst = 0.519)	−146.23	0.00	0.66
5	Smooth (Hurst = 0.934) with additional covariate	−213.51	0.31	0.54

Table 1: A summary of the I-prior models fitted on the Tecator data set.